

(19)日本国特許庁 (J P)

(12) 公表特許公報 (A)

(11)特許出願公表番号

特表2003-504753

(P2003-504753A)

(43)公表日 平成15年2月4日(2003.2.4)

(51)Int.Cl. ⁷	識別記号	F I	テーマコード(参考)
G 0 6 F 9/46	3 4 0	G 0 6 F 9/46	3 4 0 B 5 B 0 7 6
9/44	5 3 0	9/44	5 3 0 S 5 B 0 9 8
9/54		H 0 4 N 5/44	Z 5 C 0 2 5
H 0 4 N 5/44		G 0 6 F 9/06	6 4 0 C

審査請求 未請求 予備審査請求 有 (全 69 頁)

(21)出願番号 特願2001-510083(P2001-510083)
(86)(22)出願日 平成12年7月13日(2000.7.13)
(85)翻訳文提出日 平成14年1月11日(2002.1.11)
(86)国際出願番号 PCT/US00/19167
(87)国際公開番号 WO01/004743
(87)国際公開日 平成13年1月18日(2001.1.18)
(31)優先権主張番号 60/143,428
(32)優先日 平成11年7月13日(1999.7.13)
(33)優先権主張国 米国(US)
(31)優先権主張番号 09/465,994
(32)優先日 平成11年12月16日(1999.12.16)
(33)優先権主張国 米国(US)

(71)出願人 サン・マイクロシステムズ・インコーポレイテッド
Sun Microsystems, Inc.
アメリカ合衆国 カリフォルニア 95054,
サンタ クララ, ネットワーク サークル 4150
(72)発明者 カルダー・パートレイ・エイチ.
アメリカ合衆国 カリフォルニア州95135
サン・ホセ, カラッチ・レーン, 4058
(74)代理人 特許業務法人 明成国際特許事務所

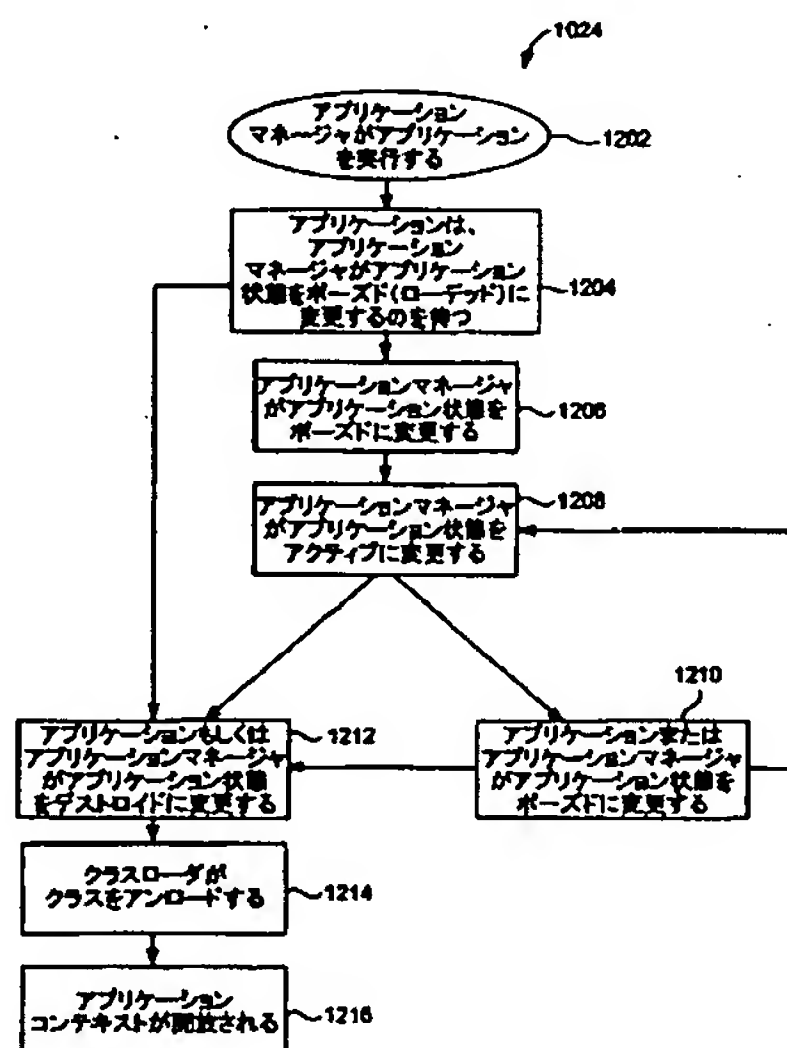
最終頁に続く

(54)【発明の名称】 アプリケーションライフサイクルに従ってアプリケーションを管理するための方法および装置

(57)【要約】

【課題】

【解決手段】 アプリケーションライフサイクルに従ってアプリケーションの実行を管理するための方法および装置を開示する。アプリケーションライフサイクルは、アプリケーションマネージャがアプリケーションを複数の状態の1つに入らせることを可能とする一連のコマンドを通してアプリケーションマネージャによって管理される。さらに、アプリケーションは、アプリケーションマネージャと通信し、アプリケーションがアプリケーションマネージャによって要求されたように状態を変更できないことを示したり、アプリケーションマネージャがアプリケーションの状態をある特定の状態へ変更することを要求したりすることができる。



【特許請求の範囲】

【請求項1】 アプリケーションライフサイクルに従ってアプリケーションの実行を管理するためのコンピュータプログラム製品であって、

コンピュータ読み取り可能な命令を格納するコンピュータ読み取り可能な媒体を備え、

前記コンピュータ読み取り可能な命令は、

第1の状態から第2の状態へのアプリケーションの状態変更をアプリケーションマネージャが開始することを、前記アプリケーションが要求することを示す状態変更要求を、前記アプリケーションから受信するための命令と、

前記第2の状態が特定の一連のルールに従って許容可能な状態である場合には、前記状態変更要求に応じて前記アプリケーションの前記状態変更を開始するための命令と、

を含む、コンピュータプログラム製品。

【請求項2】 請求項1記載のコンピュータプログラム製品であって、

前記第2の状態は、前記アプリケーションが現在実行中であることを示すアクティブ状態である、コンピュータプログラム製品。

【請求項3】 アプリケーションライフサイクルに従ってアプリケーションの実行を管理するためのコンピュータプログラム製品であって、

コンピュータ読み取り可能な命令を格納するコンピュータ読み取り可能な媒体を備え、

前記コンピュータ読み取り可能な命令は、

新たなサービスが選択されたことを示す信号を受信するための命令と、

前記新たなサービスが選択されたときに、前記アプリケーションがアクティブ状態に入るように前記アプリケーションの実行を開始するための命令と、

前記アプリケーションが前記アクティブ状態から一時停止状態に入るように、前記アプリケーションの実行を一時停止するための命令と、

実行を再開して前記一時停止状態から前記アクティブ状態に入るように前記アプリケーションが要求していることを示す再開要求を受信するための命令と、

前記再開要求が前記アプリケーションから受信されたときに、前記アプリケー

ションが前記一時停止状態から前記アクティブ状態に入るように前記アプリケーションの実行を開始するための命令と、
を含む、コンピュータプログラム製品。

【請求項4】 アプリケーションライフサイクルに従って複数のアプリケーションの実行を管理するためのコンピュータプログラム製品であって、

コンピュータ読み取り可能な命令を格納するコンピュータ読み取り可能な媒体を備え、

前記コンピュータ読み取り可能な命令は、

前記複数のアプリケーションがアクティブ状態に入るように、前記複数のアプリケーションの各々の実行を開始するための命令と、

前記複数のアプリケーションのうちの1つが前記アクティブ状態から一時停止状態に入るように、前記1つのアプリケーションの実行を一時停止するための命令と、

前記複数のアプリケーションのうちの1つ以上が実行を再開して前記一時停止状態から前記アクティブ状態に入るよう要求していることを示す再開要求を、前記1つ以上のアプリケーションから受信するための命令と、

前記一時停止状態から前記アクティブ状態へ移行させるために、前記1つ以上のアプリケーションうちの1つを選択するための命令と、

前記選択されたアプリケーションから前記再開要求を受信したときに、前記選択されたアプリケーションが前記一時停止状態から前記アクティブ状態に入るように前記選択されたアプリケーションの実行を開始させるための命令と、
を含む、コンピュータプログラム製品。

【請求項5】 アプリケーションライフサイクルに従ってアプリケーションの実行を管理するためのコンピュータプログラム製品であって、

コンピュータ読み取り可能な命令を格納するコンピュータ読み取り可能な媒体を備え、

前記コンピュータ読み取り可能な命令は、

アプリケーションが前記アプリケーションの状態を第1の状態から第2の状態へ変更するよう1回目の要求を行うための命令と、

前記アプリケーションが前記アプリケーションの状態を前記第1の状態から前記第2の状態へ変更したか否かを決定するための命令と、

前記アプリケーションが前記アプリケーションの状態を前記第1の状態から前記第2の状態へ変更していないと決定されるとともに所定の条件が満たされている場合には、前記アプリケーションが前記アプリケーションの状態を前記第1の状態から前記第2の状態へ変更するよう2回目の要求を行うための命令と、を含む、コンピュータプログラム製品。

【請求項6】 請求項5記載のコンピュータプログラム製品であって、

前記所定の条件は、特定の時間が経過したこと、あるいは前記要求された状態変更を前記アプリケーションが実行できるようになったこと、を表す、コンピュータプログラム製品。

【請求項7】 請求項5記載のコンピュータプログラム製品であって、

前記アプリケーションによって状態変更例外が生成された場合には、前記アプリケーションは前記アプリケーションの状態を変更していないものと決定される、コンピュータプログラム製品。

【請求項8】 請求項5記載のコンピュータプログラム製品であって、

前記アプリケーションが前記要求された状態変更を拒絶した場合には、前記アプリケーションが前記アプリケーションの状態を変更していないものと決定される、コンピュータプログラム製品。

【請求項9】 請求項5記載のコンピュータプログラム製品であって、

前記アプリケーションが前記要求された状態変更を実行できない場合には、前記アプリケーションが前記アプリケーションの状態を変更していないものと決定される、コンピュータプログラム製品。

【請求項10】 アプリケーションライフサイクルに従ってアプリケーションの実行を管理するためのコンピュータプログラム製品であって、

コンピュータ読み取り可能な命令を格納するコンピュータ読み取り可能な媒体を備え、

前記コンピュータ読み取り可能な命令は、

アプリケーションが前記アプリケーションの状態を第1の状態から第2の状態

へ変更するよう要求を行うための命令と、

前記アプリケーションが前記アプリケーションの状態を前記第1の状態から前記第2の状態へ変更したか否かを決定するための命令と、

前記アプリケーションが前記アプリケーションの状態を前記第1の状態から前記第2の状態へ変更していないと決定された場合には、前記アプリケーションが前記アプリケーションの状態を前記第1の状態から第3の状態へ変更するよう要求を行うための命令と、

を含むコンピュータプログラム製品。

【請求項11】 請求項10記載のコンピュータプログラム製品であって、

前記第1の状態は、前記アプリケーションが現在実行中であることを示すアクティブ状態であり、

前記第2の状態は、前記アプリケーションの前記実行が終了したことを示す抹消状態であり、

前記第3の状態は、前記アプリケーションが実行を再開できるように前記アプリケーションの実行が一時停止されたことを示す一時停止状態である、コンピュータプログラム製品。

【請求項12】 請求項10記載のコンピュータプログラム製品であって、

前記アプリケーションによって状態変更例外が生成された場合には、前記アプリケーションが前記アプリケーションの状態を変更していないものと決定される、コンピュータプログラム製品。

【請求項13】 請求項10記載のコンピュータプログラム製品であって、

前記アプリケーションが前記要求された状態変更を拒絶した場合には、前記アプリケーションが前記アプリケーションの状態を変更していないものと決定される、コンピュータプログラム製品。

【請求項14】 請求項10記載のコンピュータプログラム製品であって、

前記アプリケーションが前記要求された状態変更を実行できない場合には、前記アプリケーションが前記アプリケーションの状態を変更していないものと決定される、コンピュータプログラム製品。

【請求項15】 アプリケーションライフサイクルに従ってアプリケーショ

ンの実行を管理するためのコンピュータプログラム製品であって、

コンピュータ読み取り可能な命令を格納するコンピュータ読み取り可能な媒体を備え、

前記コンピュータ読み取り可能な命令は、

第1のアプリケーションが前記第1のアプリケーションの状態を第1の状態から第2の状態へ変更するよう要求を行うための命令と、

前記第1のアプリケーションが前記第1のアプリケーションの状態を前記第1の状態から前記第2の状態へ変更したか否かを決定するための命令と、

前記第1のアプリケーションが前記第1のアプリケーションの状態を前記第1の状態から前記第2の状態へ変更していないと決定された場合には、第2のアプリケーションが前記第2のアプリケーションの状態を前記第1の状態から前記第2の状態へ変更するよう要求を行うための命令と、
を含むコンピュータプログラム製品。

【請求項16】 請求項15記載のコンピュータプログラム製品であって、

前記第1の状態は、アクティブ状態、一時停止状態、もしくはロード済み状態であり、

前記第2の状態は、前記アプリケーションが終了されることを示す抹消状態である、コンピュータプログラム製品。

【請求項17】 請求項15記載のコンピュータプログラム製品であって、

前記第1のアプリケーションによって状態変更例外が生成された場合には、前記第1のアプリケーションが前記第1のアプリケーションの状態を変更していないものと決定される、コンピュータプログラム製品。

【請求項18】 請求項17記載のコンピュータプログラム製品であって、

前記第2の状態は、前記関連するアプリケーションが実行中であることを示すアクティブ状態であり、

前記状態変更例外は、前記第1のアプリケーションが前記第1のアプリケーションを一時停止状態もしくは抹消状態に入らせた場合に、前記第1のアプリケーションによって生成される、コンピュータプログラム製品。

【請求項19】 請求項15記載のコンピュータプログラム製品であって、

前記第1のアプリケーションが前記要求された状態変更を拒絶した場合には、前記第1のアプリケーションが前記第1のアプリケーションの状態を変更していないものと決定される、コンピュータプログラム製品。

【請求項20】 請求項15記載のコンピュータプログラム製品であって、前記第1のアプリケーションが前記要求された状態変更を実行できない際には、前記第1のアプリケーションが前記第1のアプリケーションの状態を変更していないものと決定される、コンピュータプログラム製品。

【請求項21】 アプリケーションライフサイクルに従ってアプリケーションの実行を管理するためのシステムであって、

1つ以上のルールと、

アプリケーションライフサイクルに従って1つ以上のアプリケーションを実行することができるアプリケーションマネージャであって、前記アプリケーションの各々が1つ以上の関連する所定のコマンドに応じて複数の状態の1つに入ることとを可能とするとともに、前記1つ以上のルールに従って前記所定のコマンドの1つを実行させるために選択することが可能なアプリケーションマネージャと、を備えるシステム。

【請求項22】 請求項21記載の管理システムであって、さらに、

前記アプリケーションマネージャに接続されるとともに、データストリームを受信可能な信号モニタを備え、

前記信号モニタは、アプリケーションが前記データストリーム内に存在するかどうかを決定するとともに、前記アプリケーションに関する情報を前記アプリケーションマネージャと通信するように構成されている、システム。

【請求項23】 請求項21記載の管理システムであって、

前記アプリケーションマネージャは、前記アプリケーションの各々のためのアプリケーションコンテキストを格納するよう構成されており、

前記アプリケーションコンテキストは、前記複数の状態のうちの現在の状態を識別する、システム。

【請求項24】 請求項23記載の管理システムであって、

前記複数の状態のうちの前記現在の状態は、前記アプリケーションマネージャ

に対して前記関連するアプリケーションによって識別される、システム。

【請求項25】 請求項23記載の管理システムであって、

前記アプリケーションコンテキストは、さらに、前記アプリケーションに関する1つ以上のクラスをロードできるクラスローダを識別する、システム。

【請求項26】 請求項23記載の管理システムであって、

前記アプリケーションコンテキストは、さらに、表示されるディスプレイ情報を含むディスプレイコンテキストを識別する、システム。

【請求項27】 請求項23記載の管理システムであって、

前記アプリケーションコンテキストは、さらに、前記関連するアプリケーションが前記アプリケーションマネージャと通信することを可能とするアプリケーション環境オブジェクトを識別する、システム。

【請求項28】 請求項23記載の管理システムであって、

前記アプリケーションコンテキストは、さらに、前記アプリケーションのランタイム環境に関するプロパティの、前記関連するアプリケーションによる検索を可能とするアプリケーション環境オブジェクトを識別する、システム。

【請求項29】 請求項23記載の管理システムであって、

前記アプリケーションコンテキストは、さらに、前記複数の状態の1つへの状態変更を、前記関連するアプリケーションが通信することを可能とするアプリケーション環境オブジェクトを識別する、システム。

【請求項30】 請求項23記載の管理システムであって、

前記アプリケーションコンテキストは、さらに、前記アプリケーションマネージャが前記アプリケーションの前記現在の状態を一時停止状態からアクティブ状態へ変更するように、前記関連するアプリケーションが要求を行うことを可能とするアプリケーション環境オブジェクトを識別する、システム。

【請求項31】 請求項21記載の管理システムであって、さらに、

前記アプリケーションマネージャに接続されるとともに、前記アプリケーションの各々のためのディスプレイコンテキストを管理するよう構成されたディスプレイマネージャを備え、

前記ディスプレイコンテキストは、前記ディスプレイコンテキストが可視であ

るときには第1の状態にあり、前記ディスプレイコンテキストが不可視のときには第2の状態にある、システム。

【請求項32】 請求項31記載の管理システムであって、

前記ディスプレイコンテキストは、前記関連するアプリケーションがアクティブ状態にあるときには前記第1の状態にあり、

前記関連するアプリケーションが一時停止状態にあるときには前記第2の状態にある、システム。

【請求項33】 請求項31記載の管理システムであって、

前記ディスプレイコンテキストの前記状態は、前記アプリケーションマネージャが従う前記1つ以上のルールによって決定される、システム。

【請求項34】 アプリケーションライフサイクルに従ってアプリケーションの実行を管理するためのデジタルTV受信装置であって、

プロセッサと、

コンピュータ読み取り可能な命令を格納するメモリと、

を備え、

前記コンピュータ読み取り可能な命令は、

アプリケーションがデータストリーム内に存在するか否かを、前記データストリームから決定するための命令と、

前記アプリケーションが前記データストリーム内に存在することが決定された場合に、前記アプリケーションをロードするための命令と、

複数の状態を含むアプリケーションライフサイクルに従って前記アプリケーションを実行するための命令と、

を含む、受信装置。

【請求項35】 請求項34記載のデジタルTV受信装置であって、

前記アプリケーションを実行するための前記命令は、

アプリケーションマネージャから可視であるとともに、前記アプリケーションマネージャが、前記アプリケーションを前記複数の状態のうちの1つから前記複数の状態のうちの他の1つへ変更させることを可能とするよう構成された第1のインターフェースと、

前記アプリケーションから可視であるとともに、前記複数の状態のうちの第1の組から前記複数の状態のうちの第2の組への前記アプリケーションの状態変更を、前記アプリケーションが前記アプリケーションマネージャと通信することを可能とするよう構成されている第2のインターフェースと、
を備える受信装置。

【請求項36】 請求項35記載のデジタルTV受信装置であって、
前記複数の状態の前記第2の組は、前記アプリケーションが一時停止されたことを示す一時停止状態と、前記アプリケーションが終了されたことを示す抹消状態とを含む、受信装置。

【請求項37】 請求項34記載のデジタルTV受信装置であって、
前記アプリケーションを実行するための前記命令は、
アプリケーションマネージャから可視であるとともに、前記アプリケーションマネージャによって、前記アプリケーションが前記アプリケーションの状態を前記複数の状態のうちの1つから前記複数の状態のうちの他の1つへ変更することを可能とするように構成された第1のインターフェースと、
前記アプリケーションから可視であるとともに、前記アプリケーションマネージャによって前記アプリケーションの前記状態が前記複数の状態のうちの第1の状態へ変更されることを、前記アプリケーションが要求を行うことを可能とするように構成されている第2のインターフェースと、
を備える受信装置。

【請求項38】 請求項37記載のデジタルTV受信装置であって、さらに、
前記アプリケーションの前記状態を前記複数の状態のうちの第2の状態から前記複数の状態のうちの前記第1の状態へ変更するための命令を含む、受信装置。

【請求項39】 請求項38記載のデジタルTV受信装置であって、
前記第1の状態はアクティブ状態であり、
前記第2の状態は一時停止状態である、受信装置。

【請求項40】 請求項34記載のデジタルTV受信装置であって、
前記アプリケーションを実行するための前記命令は、

アプリケーションマネージャから可視であるとともに、前記アプリケーションマネージャによって、前記アプリケーションが前記アプリケーションの状態を前記複数の状態の1つから前記複数の状態の他の1つへ変更させることを可能とするよう構成された第1のインターフェースと、

前記アプリケーションから可視であるとともに、前記アプリケーションマネージャによって要求されたように前記アプリケーションが前記アプリケーションの状態を変更できないことを、前記アプリケーションが前記アプリケーションマネージャと通信することを可能とするよう構成されている第2のインターフェースと、
を備える受信装置。

【請求項41】 請求項40記載のデジタルTV受信装置であって、さらに

前記アプリケーションマネージャによって要求された前記アプリケーションの状態変更を前記アプリケーションが実行できないことを示す状態変更例外を、前記アプリケーションが生成することを可能とする命令を含む、受信装置。

【請求項42】 請求項40記載のデジタルTV受信装置であって、さらに

前記アプリケーションマネージャによって要求されたような前記アプリケーションの状態変更を前記アプリケーションが望まないことを示す状態変更例外を、前記アプリケーションが生成することを可能とする命令を含む、受信装置。

【請求項43】 請求項36記載のデジタルTV受信装置であって、さらに

前記アプリケーションが終了されたときに、前記アプリケーションに関連するメモリを開放するための命令を含む、受信装置。

【請求項44】 請求項34記載のデジタルTV受信装置であって、さらに

前記アプリケーションに関するクラスローダであって、前記アプリケーションに関する1つ以上のクラスをロードするよう構成されているクラスローダを生成するための命令と、

前記アプリケーションに関する前記クラスをロードするために前記クラスローダを用いるための命令と、

前記クラスローダによってロードされた前記クラスを用いて、前記アプリケーションをインスタンス化するための命令と、
を備える受信装置。

【請求項45】 請求項44記載のデジタルTV受信装置であって、さらに

前記アプリケーションが終了されるときに、前記アプリケーションに関連する前記クラスをアンロードするための命令を含む、受信装置。

【請求項46】 請求項45記載のデジタルTV受信装置であって、

前記クラスをアンロードするための前記命令は、前記クラスローダを参照解除するための命令を含む、受信装置。

【請求項47】 デジタルTV受信装置環境においてアプリケーションの実行を管理するアプリケーションマネージャのための状態マシンであって、

前記アプリケーションがロードされたロード済み状態と、

前記ロード済み状態から前記一時停止状態へ遷移するために初期化され、前記アプリケーションが一時停止されている一時停止状態と、

前記一時停止状態から前記アクティブ状態へ遷移するために起動され、前記アプリケーションが実行中であるアクティブ状態と、

前記アクティブ状態ないし前記一時停止状態のいずれかから前記抹消状態へ遷移するよう終了され、前記アプリケーションが抹消される抹消状態と、
を備える状態マシン。

【請求項48】 請求項47記載の状態マシンであって、

前記アプリケーションは、前記ロード済み状態の際に終了されるときに、前記ロード済み状態から前記抹消状態へ遷移可能である、状態マシン。

【請求項49】 請求項48記載の状態マシンであって、

前記アプリケーションマネージャまたは前記アプリケーションのいずれかが、前記抹消状態への前記遷移を開始することができる、状態マシン。

【請求項50】 請求項47記載の状態マシンであって、

前記アプリケーションは、前記アプリケーションが一時停止されているときに、前記アクティブ状態から前記一時停止状態へ遷移可能である、状態マシン。

【請求項51】 請求項50記載の状態マシンであって、
前記アプリケーションマネージャまたは前記アプリケーションのいずれかが、前記アクティブ状態から前記一時停止状態への前記遷移を開始可能である、状態マシン。

【請求項52】 請求項47記載の状態マシンであって、
前記アプリケーションマネージャのみが、前記アプリケーションを起動することによって前記一時停止状態から前記アクティブ状態への前記遷移を開始可能である、状態マシン。

【請求項53】 請求項47記載の状態マシンであって、
前記状態マシンの前記状態は、共にアプリケーションライフサイクルを形成する、状態マシン。

【請求項54】 アプリケーションライフサイクルに従ってアプリケーションの実行を管理するためのコンピュータプログラム製品であって、

コンピュータ読み取り可能な命令を格納するコンピュータ読み取り可能な媒体を備え、

前記コンピュータ読み取り可能な命令は、

前記アプリケーションがロード済み状態に入るように前記アプリケーションをロードするための命令と、

前記アプリケーションが前記ロード済み状態にあるときに前記アプリケーションが一時停止状態に入るように前記アプリケーションを初期化するための命令と、

前記アプリケーションが前記一時停止状態にあるときに前記アプリケーションがアクティブ状態に入るように前記アプリケーションの実行を開始するための命令と、

前記アプリケーションが前記ロード済み状態、前記一時停止状態、もしくは前記アクティブ状態にあるときに、前記アプリケーションが前記抹消状態に入るように前記アプリケーションの前記実行を終了するための命令と、

を備える、コンピュータプログラム製品。

【請求項55】 請求項54記載のコンピュータプログラム製品であって、さらに、

前記アプリケーションが前記アクティブ状態にあるときに、前記アプリケーションが前記一時停止状態に入るように前記アプリケーションを一時停止するための命令を含む、コンピュータプログラム製品。

【請求項56】 請求項54記載のコンピュータプログラム製品であって、前記アプリケーションが前記一時停止状態にあるときには、前記アプリケーションの実行を開始するための前記命令が、前記アプリケーションによって呼び出し不可能である、コンピュータプログラム製品。

【請求項57】 請求項54記載のコンピュータプログラム製品であって、前記アプリケーションが前記一時停止状態にあるときには、前記アプリケーションの実行を開始するための前記命令が、前記アプリケーションの外部のプロセスによってのみ呼び出し可能である、コンピュータプログラム製品。

【請求項58】 請求項55記載のコンピュータプログラム製品であって、前記アプリケーションが前記アクティブ状態にあるときには、前記アプリケーションの前記実行を一時停止するための前記命令が、前記アプリケーションもしくは前記アプリケーションの外部のプロセスによって呼び出し可能である、コンピュータプログラム製品。

【請求項59】 請求項54記載のコンピュータプログラム製品であって、前記アプリケーションを終了するための前記命令は、前記アプリケーションもしくは前記アプリケーションの外部のプロセスによって実行可能である、コンピュータプログラム製品。

【請求項60】 アプリケーションライフサイクルに従ってアプリケーションの実行を管理するためのコンピュータプログラム製品であって、

コンピュータ読み取り可能な命令を格納するコンピュータ読み取り可能な媒体を備え、

前記コンピュータ読み取り可能な命令は、

前記アプリケーションが一時停止状態に入るように前記アプリケーションを初

期化するための命令と、

前記アプリケーションがアクティブ状態に入るように前記アプリケーションの実行を開始するための命令と、

前記アプリケーションが前記一時停止状態に入るように前記アプリケーション前記実行を一時停止するための命令と、

前記アプリケーションが抹消状態に入るように前記アプリケーションを終了するための命令と、

を含むコンピュータプログラム製品。

【請求項61】 アプリケーションライフサイクルに従ってアプリケーションの実行を管理するためのコンピュータプログラム製品であって、

コンピュータ読み取り可能な命令を格納するコンピュータ読み取り可能な媒体を備え、

前記コンピュータ読み取り可能な命令は、

前記アプリケーションがアクティブ状態に入るように前記アプリケーションの実行を開始するための命令と、

前記アプリケーションが前記一時停止状態に入るように前記アプリケーションの前記実行を一時停止するための命令と、

所定の条件が満たされた場合には、前記アプリケーションが抹消状態に入るように前記アプリケーションの前記実行を条件付きで終了するための命令と、

前記所定の条件が満たされない場合には、前記アプリケーションが前記抹消状態に入るように前記アプリケーションの前記実行を無条件に終了するための命令と、

を含むコンピュータプログラム製品。

【請求項62】 請求項61記載のコンピュータプログラム製品であって、前記所定の条件は、前記アプリケーションから受信される信号である、コンピュータプログラム製品。

【請求項63】 請求項61記載のコンピュータプログラム製品であって、前記所定の条件は、特定の期間内に前記アプリケーションから信号が受信されないことである、コンピュータプログラム製品。

【請求項64】 請求項61記載のコンピュータプログラム製品であって、
さらに、

前記所定の条件が満たされない場合には、前記アプリケーションによって生成された状態変更例外を無視するための命令を備え、

前記状態変更例外は、前記アプリケーションが終了を望まないことを示している、コンピュータプログラム製品。

【請求項65】 アプリケーションライフサイクルに従ってアプリケーションの実行を管理するためのコンピュータプログラム製品であって、

コンピュータ読み取り可能な命令を格納するコンピュータ読み取り可能な媒体を備え、

前記コンピュータ読み取り可能な命令は、

前記アプリケーションがアクティブ状態に入るように前記アプリケーションの実行を開始するための命令と、

前記アプリケーションが前記一時停止状態に入るように前記アプリケーションの前記実行を一時停止するための命令と、

前記アプリケーションが抹消状態に入るように前記アプリケーションを終了するための命令と、

前記アプリケーションの実行を開始するための前記命令、前記アプリケーションの前記実行を一時停止するための前記命令、および前記アプリケーションを終了するための前記命令の実行を、前記アプリケーション以外のプロセスが開始することを可能とする一連の命令を含むインターフェースと、
を備えるコンピュータプログラム製品。

【請求項66】 請求項65記載のコンピュータプログラム製品であって、
前記インターフェースは、前記アプリケーションを終了するための前記命令を呼び出すよう構成されたスタブを含み、

前記スタブは、パラメータを受け入れることが可能であり、

前記パラメータは、前記アプリケーションの終了が、前記パラメータが第1の状態にあるときには無条件であることを示し、前記パラメータが第2の状態にあるときには条件付きであることを示す、コンピュータプログラム製品。

【請求項67】 アプリケーションライフサイクルに従ってアプリケーションの実行を管理するためのコンピュータプログラム製品であって、

コンピュータ読み取り可能な命令を格納するコンピュータ読み取り可能な媒体を備え、

前記コンピュータ読み取り可能な命令は、

前記アプリケーションが終了することを決定して、前記アプリケーションがロード済み状態、一時停止状態、もしくはアクティブ状態から抹消状態に入ったことを通信するための命令と、

前記アプリケーションが前記アプリケーションの実行を一時停止することを決定して、前記アプリケーションが前記アクティブ状態から前記一時停止状態に入ったことを通信するための命令と、

を含む、コンピュータプログラム製品。

【請求項68】 請求項67記載のコンピュータプログラム製品であって、さらに、

前記アプリケーションが、実行を再開して前記一時停止状態から前記アクティブ状態に入ることを要求していることを通信するための命令を備える、コンピュータプログラム製品。

【請求項69】 請求項67記載のコンピュータプログラム製品であって、さらに、

前記アプリケーションのランタイム環境に関する情報を取得するための命令を備える、コンピュータプログラム製品。

【請求項70】 請求項67記載のコンピュータプログラム製品であって、さらに、

前記アプリケーションが終了することを決定したことを通信するための前記命令、および、前記アプリケーションが実行を一時停止することを決定したことを通信するための前記命令の実行を、前記アプリケーションが開始することを可能とする一連の命令を含むインターフェースを備える、コンピュータプログラム製品。

【請求項71】 請求項68記載のコンピュータプログラム製品であって、

さらに、

前記アプリケーションが終了することを決定したことを通信するための前記命令、前記アプリケーションが実行を一時停止することを決定したことを通信するための前記命令、および前記アプリケーションが実行を再開して前記一時停止状態から前記アクティブ状態に入るよう求めていることを通信するための前記命令の実行を、前記アプリケーションが開始することを可能とする一連の命令を含む、インターフェースを備える、コンピュータプログラム製品。

【請求項72】 アプリケーションライフサイクルに従ってアプリケーションの実行を管理するためのコンピュータプログラム製品であって、

コンピュータ読み取り可能な命令を格納するコンピュータ読み取り可能な媒体を備え、

前記コンピュータ読み取り可能な命令は、

前記アプリケーションがアクティブ状態に入るように前記アプリケーションの実行を開始するための命令であって、前記アプリケーションによって呼び出し不可能な命令と、

前記アプリケーションが一時停止状態に入るように前記アプリケーションの前記実行を一時停止するための命令と、

前記アプリケーションが実行を再開して前記一時停止状態から前記アクティブ状態に入ることを求めていることを通信するための命令と、を含むコンピュータプログラム製品。

【請求項73】 請求項72記載のコンピュータプログラム製品であって、さらに、

前記アプリケーションが実行を一時停止することを決定して、前記アクティブ状態から前記一時停止状態に入ったことを通信するための命令を備える、コンピュータプログラム製品。

【請求項74】 請求項72記載のコンピュータプログラム製品であって、さらに、

前記アプリケーションが抹消状態に入るように前記アプリケーションを終了するための命令を備える、コンピュータプログラム製品。

【請求項75】 請求項74記載のコンピュータプログラム製品であって、
さらに、

前記アプリケーションが終了することを決定して、前記抹消状態に入ったこと
を通信するための命令を備える、コンピュータプログラム製品。

【発明の詳細な説明】**【0001】****【発明の属する技術分野】**

本発明は、概して、コンピュータソフトウェアに関する。より詳細には、本発明は、アプリケーションの実行を管理するための方法および装置に関する。さらに、本発明は、ソフトウェアアプリケーションのためのアプリケーションライフサイクル設計を実装するための方法および装置に関する。

【0002】**【従来の技術】**

デジタルテレビの革命は、放送テレビの歴史において最も重要な事柄の一つである。デジタルテレビの出現により、衛星、ケーブル、地上テレビチャンネルを介して、高速のデータ伝送が可能となっている。デジタルテレビは、映像と音声の質を著しく改善するばかりでなく、より多くのチャンネルを提供する。最も重要なことは、デジタルテレビの登場が、真の対話型TV時代の幕開けとなることである。たとえば、デジタル受信装置は、簡単な双方向クイズショーから、インターネット、テレビおよびウェブ型コンテンツの混合まで様々な高度サービスをユーザに提供できるようになるだろう。デジタルテレビ市場が成長するにつれ、コンテンツ開発業者は、電子プログラミングガイド、ビデオ・オン・デマンド、高度放送など次世代の対話型TVサービスを確立するための基盤として、機能が豊富で、コスト効率がよく、信頼性の高いソフトウェアプラットフォームを捜し求めている。

【0003】

Java（登録商標）は、商用のオブジェクト指向言語の草分けであり、JAVAVM（商標）プラットフォームをサポートする任意の互換デバイス上で動作可能な高移植性言語として開発された。たとえば、Java（登録商標）は、すべての主要ウェブブラウザに組み込まれている。そのため、Java（登録商標）は、任意のウェブ使用可能なコンピュータ上で、そのコンピュータのウェブブラウザによって動作する。したがって、Java（登録商標）は、セットトップボックスおよびデジタルテレビのためのソフトウェアプラットフォームとして幅広い

展望を提供する。

【0004】

オブジェクト指向のプログラミングにおいては、コードとデータがオブジェクトに組み込まれる。各オブジェクトは、そのクラスによって定義される。クラスは、オブジェクトの属性と振る舞いを決定する。言い換えると、オブジェクトは、クラスの個々のインスタンスである。

【0005】

デスクトップ環境では、ロードされたアプリケーション各々に関する一部のリソース（たとえば、クラスおよびオブジェクト）は、メモリに比較的制限が少ないために、頻繁に開放される必要もなく、厳密に監視される必要もない。しかしながら、組み込みシステムの環境、特にデジタルテレビの分野においては、メモリが貴重なリソースとなる。さらに、対話型デジタルテレビ環境では、複数のアプリケーションを動作させることが一般となるだろう。デジタルテレビサービスは、音声と、映像と、1つ以上のアプリケーションとからなるだろう。たとえば、テレビの視聴者がチャンネルを変えると、そのチャンネルによって提供される関連のサービスもしくはプログラムの各々は、複数のクラスがロードされることを必要とする可能性が高い。結果として、アプリケーションとそれに関連するクラスに、メモリが頻繁に割り当てられ、限られた量のメモリが消費されることになる。メモリが消費されると、もはやアプリケーションの動作は不可能となる。エラーの際にセットトップボックスをリブートすることは望ましくないため、これは特に重大なことである。

【0006】

JAVAプラットフォームは現在、数多くのアプリケーションモデルを定義しており、それぞれが各自のライフサイクルを持っている。一般に、これらのアプリケーションライフサイクルモデルは、JAVAプラットフォーム上での特定の問題に対処するために設計されてきた。たとえば、アプレットは、ウェブページの実行可能なコンテンツにサポートを提供するよう設計された。しかしながら、既存のアプリケーションライフサイクルモデルで、テレビ受信装置のようにメモリの限られているシステムに特有の要求に完全に対応したものはない。たとえば

、アプレットに関するクラスがロードされても、そのクラスオブジェクトはメモリから除去されない。さらに、アプレットの実行がいつ終了するかを決定することは不可能である。

【0007】

上述のことから、アプリケーションライフサイクルが、テレビ受信装置に特有の要求に対応するよう設計されれば有益であろう。さらに、アプリケーションライフサイクルに従ってアプリケーションのロードおよび実行を管理するための機構が設計されることが望ましい。

【0008】

【発明の概要】

本発明は、1つ以上のアプリケーションの管理を可能とする。さらに、1つ以上のアプリケーションの実行は、アプリケーションライフサイクルに従って管理されてもよい。これは、部分的には、アプリケーションの状態の変更を開始し監視することができるアプリケーションマネージャを用いることによって実現される。このように、様々なプラットフォーム上で一貫した方法によってアプリケーションを実行することが可能である。

【0009】

本発明の一態様によると、アプリケーションマネージャは、アプリケーションをロードし、アプリケーションが実行中に使用するリソースをアプリケーションに割り当てる。アプリケーションがロードされると、アプリケーションマネージャは、アプリケーションライフサイクルに従ってアプリケーションを実行する。これは、アプリケーションマネージャから可視のアプリケーションインターフェースを通して一実施形態に従って実現される。このアプリケーションインターフェースを通して、アプリケーションマネージャは、アプリケーションの様々な状態変更を開始することができる。一例として、アプリケーションマネージャは、アプリケーションがその実行を一時停止して一時停止状態 (p a u s e d s t a t e) に入ることを要求できる。別の例として、アプリケーションマネージャは、アプリケーションが一時停止状態からその実行を再開しアクティブ状態 (a c t i v e s t a t e) に入ることを要求できる。

【0010】

本発明の別の態様によると、アプリケーションは、状態および潜在的な状態変更に関する情報をアプリケーションマネージャに通信する。一実施形態によると、これは、部分的には、アプリケーション環境インターフェースによって実現される。このアプリケーション環境インターフェースを通して、アプリケーションは、アプリケーションマネージャがアプリケーションの状態変更を引き起こすよう要求できる。たとえば、アプリケーションは、アプリケーションマネージャがアプリケーションをアクティブ状態に入らせるよう要求できる。さらに、アプリケーションは、このアプリケーション環境インターフェースを通して、要求されたサービスをアプリケーションが実行できないことと、そのために一時停止もしくは終了したことを、その状況下で適切なときに示すことができる。アプリケーションの状態が変更されると（たとえば、アプリケーションマネージャによって引き起こされるか、アプリケーションによって実行されると）、アプリケーションは、この状態変更をアプリケーションマネージャに通信することができる（たとえば、アプリケーションから可視なアプリケーション環境インターフェースを通して）。

【0011】

本発明のさらに別の態様によると、アプリケーションマネージャは、条件付きもしくは無条件でアプリケーションを終了することができる。アプリケーションマネージャが無条件でアプリケーションを終了すると、アプリケーションは終了しなければならない。しかしながら、アプリケーションマネージャが条件付きでアプリケーションを終了すると、アプリケーションが終了に同意したときにのみ、アプリケーションマネージャはアプリケーションを終了することができる。このように、アプリケーションマネージャは、終了されるアプリケーションが同意した方法で、アプリケーションを終了することにより、アプリケーションに關係するリソースを開放することができる。

【0012】

本発明は、アプリケーションがアプリケーションマネージャによりアプリケーションライフサイクルに従って管理されることを可能とする。これは、アプリケ

ーションが、実行されるプラットフォームに関係なく予測できる方法で実行されることを可能とする。さらに、アプリケーションマネージャは、アプリケーション各々の現在の状態を監視するため、アプリケーションの終了時即座に各アプリケーションに関係するメモリを開放することができる。これは特に、デジタルTV受信装置のようにメモリの限られたシステムに有用である。

【0013】

【発明の実施の形態】

以下の説明では、本発明を完全に理解できるように、数多くの具体的な詳細を示している。しかしながら、これらの具体的な詳細の一部もしくはすべてがなくとも本発明が実施可能であることは、当業者には明らかであろう。その他、本発明が不必要に不明瞭とならないように周知のプロセスのステップについては詳細に説明していない。

【0014】

本明細書では、アプリケーションライフサイクルに従ってアプリケーションがロードおよび実行されることを可能とする発明が説明されている。アプリケーションが初期化され、その環境に基づいて様々な状態変更を行い、最終的に破棄（destroy）されるまでの一連のステップは、まとめてアプリケーションライフサイクルとして知られている。一実施形態によると、これは、1つ以上のアプリケーションをロードし実行を管理することができるアプリケーションマネージャを用いることによって実現される。一実施形態によると、アプリケーションライフサイクルは、2つのインターフェースを用いることによって実装される。まず、アプリケーションプログラミングインターフェースにより、アプリケーションマネージャが、アプリケーションライフサイクルに従ってアプリケーションの実行を管理することが可能となる。次に、アプリケーション環境インターフェースにより、アプリケーションが、1つの状態から別の状態への変更の要求をアプリケーションマネージャに通信したり、アプリケーションマネージャによって要求された状態変更の実行が不可能であることを示したりすることが可能となる。たとえば、アプリケーションは、アプリケーションマネージャがアプリケーションをアクティブ状態に入らせるよう要求できる。別の例として、アプリケーシ

ョンは、一時停止状態もしくは抹消状態 (destroyed state) に入ったことを示してもよい (たとえば、アプリケーションマネージャによって要求されたアクティブ状態が不可能である場合)。このように、アプリケーションが最終的には自身の状態を最も良く知っているため、アプリケーション環境インターフェースを用いることで、アプリケーションが、1つの状態から別の状態へ変更したことをアプリケーションマネージャに通信することが可能となる。

【0015】

本発明は、デジタルテレビおよびデジタルTV受信装置に即して説明されている。図1は、代表的なデジタルTV受信装置を示すブロック図である。図示されているように、信号は、アンテナ102を介して受信され、チューナモジュール104によって同調され、MPEG2転送ストリーム106を生成する。次いで、多重分離装置108は、映像ストリーム112と、音声ストリーム114と、データストリーム116とを含む暗号化MPEGストリーム110を生成する。次いで、これらの3つのストリームは、条件付きアクセスサブシステム118によって処理される。たとえば、条件付きアクセスサブシステム118は、復号情報122 (たとえば、復号アルゴリズム) だけでなく、キー管理情報120を用いてもよい。条件付きアクセスサブシステム118は、データ126と共に映像ストリーム124と音声ストリーム125とを含む復号化MPEGストリーム123を生成する。それらはすべて復号化されている。次いで、デコーダ128が、復号化MPEGストリーム123を処理し、デコードされた映像データをフレームバッファ130へ、デコードされた音声データをスピーカ132へ送る。

【0016】

Java仮想マシンは、図1に示すようなデジタルTV受信装置によって受信された情報を処理するよう本発明を実装するために用いることができるプラットフォームの1つである。より詳細には、データ126 (たとえば、放送データストリーム) が処理される際に、データ126で提供され、ダウンロードされるアプリケーションのような情報を処理することが望ましい。

【0017】

図2Aは、デジタルTV受信装置を通してロードされたアプリケーションを管

理するために、アプリケーションマネージャが実装された本発明の一実施形態を示すブロック図である。図2Aに示すように、本発明は、デジタルTV受信装置200に実装可能である。放送データストリーム202は、信号モニタ204によって受信される。信号モニタ204は、放送データストリーム202内にアプリケーションが存在するか否かを決定し、アプリケーションのロケーションなどアプリケーションに関するデータと共に、アプリケーションの有無を示す信号データ206を提供する。次いで、アプリケーションマネージャ208は、この信号データ206を用いてアプリケーションをロードして実行する。たとえば、アプリケーションマネージャ208は、特定の条件に応じてアプリケーションが適切な状態に入るように、適切なアプリケーションライフサイクルに従ってアプリケーションを実行することができる。アプリケーションライフサイクル設計の実装の1つについて、さらに詳細に以下で説明する。アプリケーションが実行されると、ディスプレイ情報が、ディスプレイコンテキスト210の形でディスプレイマネージャ212から提供される。次いで、ディスプレイマネージャは、適切なディスプレイ情報をテレビモニタ214のディスプレイに提供する。

【0018】

本発明の一実施形態によると、アプリケーションマネージャは、一連のルール216に従って動作するよう構成されている。これらのルールは、アプリケーションライフサイクルを実装するために様々な方法で指定されてよい。たとえば、アプリケーションライフサイクルは、所定のイベントが起こった際に、アプリケーションが、ロード済み状態 (loaded state)、一時停止状態、アクティブ状態、抹消状態に入ることを可能とすることができる。図2Bは、図2Aのアプリケーションマネージャが動作の際に従うことのできる代表的な一連のルールを示す図である。この代表的な一連のルール216は、4つのルールを含んでいる。第1のルール218は、アプリケーションマネージャが1つ以上のアプリケーションを管理することを指定する。しかしながら、第2のルール220は、常に1つのアプリケーションしかアクティブにできない (すなわち、実行できない) ことを指定する。さらに、第3のルール222は、常に1つのアプリケーションしか表示できないことを指定する。最後に、第4のルール224は、ア

クティブなアプリケーションのみが表示されることを指定する。このように、一連のルールは、様々な方法でアプリケーションマネージャのために設計、構成される。したがって、一連のルールは、アプリケーションライフサイクルに従って1つ以上のアプリケーションの実行を管理するよう設計されてもよい。

【0019】

図3は、図2Aのアプリケーションマネージャによって受信された信号データを格納するために用いられる代表的なデータ構造を示すブロック図である。信号データ302は、放送データストリーム内のアプリケーション存在表示304を含む。アプリケーションが存在する場合、そのファイルのロケーション306（たとえば、ディレクトリおよびファイル名）が指定される。さらに、ある特定の処理（たとえば、読み取り／書き込み）が、特定のソースおよび／または宛先に関連して実行可能もしくは不可能であることを示す許可310だけでなく、アプリケーションの作成元が認証されることを可能とする認証情報308のような情報も含まれる。

【0020】

アプリケーションマネージャ208は、数多くのアプリケーションのライフサイクルを同時に管理できる。本発明の一実施形態によると、潜在的に実行中のアプリケーション400の各々に関する情報を保持するために、アプリケーションマネージャは、アプリケーションコンテキストリスト402を管理する。図4Aは、関連するアプリケーションの管理および実行中に、アプリケーションマネージャ208によってアクセスされる代表的なアプリケーションコンテキストリスト402を示す図である。図示されているように、アプリケーションコンテキストリスト402は、1つ以上の入力項目を含み、各入力項目は、アプリケーション400の1つに関するアプリケーションコンテキスト404を識別する。さらに詳細には、アプリケーション400-A、400-B、400-C、400-Dは、それぞれ異なるアプリケーションコンテキスト404-A、404-B、404-C、404-Dを伴う4つの潜在的に異なるアプリケーションを示す。アプリケーションコンテキスト404は、アプリケーションが、アプリケーションライフサイクルに従って実行されるだけでなくロードされることも可能とする

ために、アプリケーションに関する情報を識別する。

【0021】

一実施形態によると、アプリケーションに関する情報は、アプリケーションコンテキスト404によって集積され、参照される。図4は、図4Aの代表的なアプリケーションコンテキストリスト内で識別されたアプリケーションコンテキスト404を格納するために用いられる代表的なデータ構造を示す図である。アプリケーションコンテキスト404の代表的なデータ構造は、クラスローダ、すなわちクラスをメモリにロードするために用いられるオブジェクト、を識別するクラスローダID406を含む。それゆえ、クラスローダID406は、アプリケーションマネージャが、識別されたクラスローダによってアプリケーションに関する1つ以上のクラスをロードすることと、クラスローダを参照解除することによりこれらのクラスをアンロードすることを可能とする。さらに、アプリケーションコンテキスト404の代表的なデータ構造は、図3に示したような信号データ302を識別する信号データID408を含む。それゆえ、アプリケーションマネージャ208は、認証および許可データだけでなくアプリケーションのローションを決定するために、この信号データID408を用いることができる。アプリケーションコンテキスト404の代表的なデータ構造は、さらに、図2Aのディスプレイマネージャ212がアプリケーションを表示するために用いる情報を含むディスプレイコンテキストを識別するディスプレイコンテキストID410を含む。たとえば、ディスプレイコンテキストは、サイズ、位置、可視性のデータなど、スクリーン上のアプリケーションの表示を与えるオブジェクトへの参照を含んでもよい。アプリケーションコンテキスト404の代表的なデータ構造は、さらに、アプリケーションを識別するアプリケーションID412を含む。さらに、アプリケーションがアプリケーションマネージャと通信することを可能とするアプリケーション環境オブジェクト414が識別される。一例として、アプリケーションは、別の状態へ入るという要求を通信したり、別の状態に入ってその状態変更をアプリケーションマネージャに通信（たとえば、一時停止（*paused*）もしくは抹消状態に入ったことを通信）したりしてもよい。別の例として、アプリケーションは、（たとえば、アプリケーションコンテキストに関

係して保持された) アプリケーション環境に関する情報を取得するよう求めてもよい。アプリケーション環境オブジェクト414に関しては、図8および図9を参照してさらに詳細に説明する。アプリケーションコンテキスト404の代表的なデータ構造は、さらに、現在のアプリケーション状態416を識別するために示される。それゆえ、アプリケーションマネージャ208は、各アプリケーションのアプリケーション状態の記録を保持する。

【0022】

図2Aを参照して上に説明したように、最終的に表示されるデータを管理するだけでなく、ディスプレイ装置へのアクセスを管理するために、別個のディスプレイマネージャ212が実装されてもよい。このプロセスを円滑にするために、ディスプレイマネージャは、本発明の一実施形態にしたがってディスプレイコンテキストのリストを保持する。図5は、本発明の一実施形態に従って、ディスプレイマネージャによってアクセスされるディスプレイコンテキストの代表的なリスト(ディスプレイコンテキストリスト)502を示す図である。図示されているように、ディスプレイコンテキストリスト502は、1つ以上のアプリケーション400-Aないし400-Dに関する1つ以上の入力項目を含む。ディスプレイコンテキスト502内の各入力項目は、コンテキストID504に関係し、アプリケーションの表示に関する情報を指定するディスプレイコンテキスト506を識別する。言い換えると、ディスプレイコンテキストは、アプリケーションが自身の表示のために必要とする情報を保持するオブジェクトであってもよい。それゆえ、ディスプレイマネージャ212が、アプリケーションの表示を望む場合、おそらく他のウィンドウを閉じることによって、対応するディスプレイコンテキスト内で参照される適切な「ウィンドウ」を「開く」ことができる。

【0023】

ディスプレイコンテキストは、ディスプレイコンテキストのための状態遷移図にしたがって表示されてもよい。図5Bは、ディスプレイコンテキストに関連する代表的な状態図である。図示されているように、ディスプレイコンテキストは、第1の表示状態においては可視508であり、第2の表示状態においては不可視510である。一実施形態によると、アプリケーションマネージャが従うルー

ルによって、適切な表示状態が決定される。

【0024】

図5Cは、図2Bを参照して上に説明したように、アプリケーションマネージャが従う一連の代表的なルールに従って生成された一連の代表的な表示状態を提供するテーブルを示す図である。テーブル512は、すべての可能な表示状態514と、それらに関連するアプリケーション状態516を示す。アプリケーションマネージャが従う代表的な一連のルールに従って、アクティブなアプリケーションのみが表示される。それゆえ、アプリケーションがアクティブ状態の際には、表示状態は可視である。あるいは、アプリケーションが一時停止状態の際には、表示状態は不可視である。

【0025】

アプリケーションライフサイクルは、アプリケーションが、所定の条件下で多くの状態に入ることを可能とすることにより実装される。アプリケーションマネージャおよびアプリケーションライフサイクルの実装に従って、アプリケーションがいつ状態の各々に入るかを決定する条件だけでなく、状態が変化してもよい。図6は、本発明の一実施形態に従って、アプリケーションのライフサイクルの間にアプリケーションが入る一連の状態を示す状態図である。アプリケーションマネージャがアプリケーションをロードすると、アプリケーションは、ロード済み状態602に入る。アプリケーションがロード済み状態602に入ると、アプリケーションは、アプリケーションマネージャによる初期化の後に一時停止状態604に入ってもよい。アプリケーションに一時停止状態604からアクティブ状態606へ状態変更させることができるのは、アプリケーションマネージャのみである。しかしながら、アプリケーションをアクティブ状態606から一時停止状態604に入らせるのは、アプリケーションマネージャとアプリケーションのいずれであってもよい。さらに、アプリケーションを終了させ、ロード済み状態602、アクティブ状態606、一時停止状態604から抹消状態608に入らせるのは、アプリケーションマネージャとアプリケーションのいずれであってもよい。

【0026】

図6を参照して上に説明したように、アプリケーションの状態は、実行中のアプリケーションもしくは実行中のアプリケーション以外のアプリケーション（たとえば、アプリケーションマネージャ）によって変更されてよい。多くのオブジェクト指向言語では、実装の詳細が外部プロセスから隠されていても、メソッド名およびパラメータが外部プロセスから可視となる（たとえば、インターフェースを通して）ように、メソッドと変数が、モジュールにグループ化されてもよい。一実施形態によると、アプリケーションおよびアプリケーションマネージャにアクセス可能なインターフェースとそれに関するメソッドは、以下でパッケージと呼ばれているものに「パッケージ化」される。各パッケージは、外部から可視なメソッド名およびパラメータを定義する関連インターフェースを含む。さらに、各パッケージは、パッケージのインターフェースで定義されたメソッドのボディと実装の詳細を含む関連ボディを有する。アプリケーションとアプリケーションマネージャは、通信のために2つの異なるインターフェースを用いてもよい。このように、アプリケーションマネージャとアプリケーションは、2つの異なるインターフェース、すなわちアプリケーションインターフェースおよびアプリケーション環境インターフェースを用いることにより、アプリケーションの状態を変更することができる。さらに、アプリケーションマネージャとアプリケーションは、状態変更もしくは潜在的な状態変更に関する情報などの情報を通信することができる。

【0027】

図7は、アプリケーションの状態を変更するために、アプリケーションのライフサイクルの間にアプリケーションマネージャ（もしくは他のプロセス）が呼び出すことのできるメソッドを識別する代表的なアプリケーションインターフェースを示す図である。アプリケーションインターフェース702は、アプリケーションマネージャが呼び出すことのできるすべてのメソッドとそれに関連するパラメータを定義する。initializeメソッド704は、アプリケーションマネージャがアプリケーションを初期化することを可能とするために利用することができる。たとえば、initializeメソッド704は、アプリケーションが自身を初期化し、ロード済み状態から一時停止状態へ入るよう信号を送る

ことができる。一実施形態によると、`initialize`メソッド704のパラメータは、アプリケーション環境オブジェクト706である。アプリケーション環境オブジェクト706に関しては、図8を参照してさらに詳細に説明する。また、アプリケーション環境オブジェクト706は、以下に説明するように、アプリケーションのランタイム環境に関する属性にアクセスするために、アプリケーションによって用いられてもよい。さらに、アプリケーション環境オブジェクト706を用いることにより、アプリケーションは、そのランタイム環境に関する属性を検索することができる。たとえば、検索可能な属性は、信号データへの参照と、サービスの出所となる環境（たとえば、音声／映像環境）への参照を含む。このように、アプリケーションは、実行に備えて自身を初期化できる。アプリケーションは、実行に先立って共有リソースを保持しないことが好ましいが、適度な時間で実行するよう準備されるべきである。`initialize`メソッド704は、一回だけ呼び出されるべきである。`initialize`メソッド704がうまく戻った後、アプリケーションは一時停止状態となる。`initialize`メソッド704がうまく戻ることができなかった場合、アプリケーションは、一時停止状態に入ることができないことを示す状態変更例外を戻す。

【0028】

アプリケーションが一時停止状態になると、そのアプリケーションは、アクティブ状態に入ってもよい。しかしながら、`start`メソッド708を用いてこの状態変更を引き起こすことができるのは、アプリケーションマネージャだけである。`start`メソッド708は、一旦呼び出されると、アプリケーションがサービス提供を開始し、アクティブ状態に入るよう信号通知する。アクティブ状態においては、アプリケーションが共有リソースを保持してもよい。様々な障害によって、サービスの開始が妨げられることがある。たとえば、障害は、一過性であったり一過性でなかったりする。一実施形態によると、アプリケーションは、これら2種類の障害を区別する。一過性の障害に対しては、状態変更例外が生成される。一過性でない障害が生じた際には、別の例外が生成されてもよいし、`done`メソッドが呼び出され、メソッドが適切に終了されてもよい。たとえば、アプリケーションが、実行のために必要なリソースにアクセスできないと決定

した場合、これは、一過性もしくは非一過性の障害として実装されてもよい。

【0029】

`pause`メソッド710は、アプリケーションマネージャによって呼び出されると、アプリケーションが実行を停止し、アクティブ状態から一時停止状態に入るよう信号通知する。一時停止状態においては、アプリケーションは実行を停止し、できる限り使用するリソースを少なくするよう努める。このように、アプリケーションは、保持している共有リソースの一部もしくはすべてを開放することができる。

【0030】

ロード済み (`loaded`)、アクティブ (`active`)、一時停止状態のいずれかからアプリケーションを終了することが望ましい。これは、`destroy`メソッド712を呼び出すことによって実現することができる。`destroy`メソッド712は、アプリケーションが終了し、抹消状態に入るよう信号通知する。抹消状態においては、アプリケーションは、すべてのリソースを開放し、設定もしくは状態の情報を保存する必要がある。

【0031】

アプリケーションマネージャがアプリケーションの抹消（たとえば、終了）を求める場合には、様々なレベルの緊急を知らせることが望ましい。一実施形態によると、アプリケーションの終了は、所定の条件が満たされた場合に実行される。一例として、所定の条件は、早期終了に同意しているアプリケーションからの信号が存在するという条件でもよい。別の例として、所定の条件は、特定の期間内にアプリケーションからの信号が受信されないという条件でもよい。これは、`destroy`メソッド712に対するパラメータ714によって実現することができる。パラメータ714は、第1の状態のときには抹消信号が条件付きであり、第2の状態のときには無条件であることを示す。それゆえ、アプリケーションマネージャは、単にアプリケーションの終了を要求する際には、抹消信号が、アプリケーションによる終了の決定を条件とすると指定してもよい。アプリケーションは、状態変更例外を生成することによって、実行の継続を求める、すなわち、抹消状態に入るよう求めないことを示してもよい。アプリケーションマネー

ジャは、アプリケーションからのこの要求を認める場合、後に再びdestroyメソッドを呼び出してもよい。あるいは、アプリケーションマネージャは、たとえば優先順位の高い（もしくは低い）他のアプリケーションを抹消するよう試みてもよい。このように、アプリケーションマネージャは、終了を選択したアプリケーションからのみ、（たとえば、次にロードされるアプリケーションによって）必要とされるリソースを得ることができる。さらに、アプリケーションマネージャは、優先順位、メモリ使用量が最大（最小）のアプリケーション、実行が開始された順序など、特定の順序でアプリケーションを抹消するよう試みてもよい。

【0032】

アプリケーションマネージャは、ある特定のアプリケーションの終了を必要とする際には、destroyメソッド712のパラメータ714を通じて抹消信号が無条件であると指定して、これを示してもよい。たとえば、アプリケーションマネージャは、アプリケーションによって保持されているリソースすべてがアプリケーションマネージャによって必要とされている場合には、抹消信号が無条件であると指定してもよい。このように、アプリケーションマネージャは、抹消信号が無条件である場合には、アプリケーションを強制的に終了し、保持されているリソースを開放してもよい。したがって、アプリケーションが、実行の継続を求めることを示す状態変更例外を生成している場合でも、抹消信号が無条件であれば、アプリケーションマネージャは、この例外を無視してもよい。上の説明では、アプリケーションの条件付き終了と無条件終了に言及しているが、別の動作が、条件付きおよび無条件で同様に実行されてもよい。

【0033】

上述のように、アプリケーション環境オブジェクトは、アプリケーションが初期化された際に、アプリケーションに渡される。さらに、アプリケーション環境オブジェクトは、内部の状態変更を信号通知する方法と共に、属性を検索する機構をアプリケーションに提供する。一実施形態によると、アプリケーション環境オブジェクトは、ロードおよび実行されているアプリケーションに利用可能なアプリケーション環境インターフェースを備える。より詳細には、アプリケーショ

ンが初期化されると、アプリケーション環境インターフェースが、アプリケーションに利用可能となる。

【0034】

図8は、アプリケーションのライフサイクルの間に、実行中のアプリケーションが呼び出すことのできるメソッドを識別する代表的なアプリケーション環境インターフェースを示す図である。以下に説明するように、アプリケーション環境インターフェース802は、アプリケーションが、その状態変更と所望の状態変更に関してアプリケーションマネージャと通信することと共に、その環境に関する情報を発見することを可能とするいくつかのメソッドを提供する。

【0035】

上述のように、アプリケーションは、ロード済み、アクティブ、一時停止状態から、抹消状態に入ることができる。さらに、アプリケーションマネージャもしくはアプリケーションは、アプリケーションを抹消状態に入らせることができる。図7を参照して説明したように、アプリケーションマネージャは、destroyメソッドを呼び出すことによりアプリケーションを抹消することができる。あるいは、アプリケーション環境インターフェース802のdestroyedメソッド804は、アプリケーションが自身を抹消状態に入れたことを信号通知することを可能とする。次いで、アプリケーションマネージャは、図7を参照して上述したように、アプリケーションインターフェースに備えられたdestroyメソッドを呼び出さずに、アプリケーション状態を抹消状態に更新する。アプリケーションは、アプリケーションマネージャによって抹消されたかのように同じ動作（たとえば、クリーンアップ、リソースの開放）を行う。これは、アプリケーションが自身を抹消状態に入らせる前に実行されることが好ましい。

【0036】

アプリケーションがアクティブ状態であるときに、アプリケーションマネージャもしくはアプリケーションは、そのアプリケーションを一時停止状態に入らせてもよい。より詳細には、アプリケーションマネージャは、図7を参照して説明したように、pauseメソッドを用いてアプリケーションを一時停止することができる。あるいは、アプリケーションが、アクティブ状態を求めないことと、

pausedメソッド806によって一時停止状態に入ったことを信号通知してもよい。

【0037】

パラメータ810を持つget propertyメソッド808は、アプリケーション環境オブジェクトから1つ以上の属性を検索する機構をアプリケーションに提供する。一例では、Java環境において、「ルートコンテナ」は、一般にユーザインターフェースのコンポーネント（たとえば、プルダウンメニュー、ボタン）がJava環境に表示可能になるように、それらを含むために用いられる。それゆえ、get propertyメソッド808は、コンポーネントを含むことができるルートコンテナを決定するために用いてもよい。別の例として、放送データストリームに関する情報（たとえば、放送データストリーム内のファイルのロケーション）が取得されてもよい。たとえば、別の属性が、サービスセッションオブジェクト（サービスコンテキスト）への参照と、基礎となるトランスポートプロトコルに固有の情報とを含んでもよい。

【0038】

アプリケーションをアクティブ状態に入らせることができるのはアプリケーションマネージャのみであるが、アプリケーションは、アクティブ状態に入ることに関心があることを示すよう望んでもよい。一実施形態によると、これは、resume requestメソッド812によって実現される。resume requestメソッド812によって、1つ以上のアプリケーションがそれぞれ、アクティブ状態に入る要求を示すことができる。しかしながら、同時に実行可能なアプリケーションの数は、アプリケーションマネージャが従う一連のルールによって制限されるだろう。たとえば、図2Bを参照して上述したように、ルールによって、一度にアクティブにできるのは1つのアプリケーションのみであると指定されてもよい。それゆえ、resume requestメソッド812は、アクティブ状態に入ることを求めるアプリケーションを決定することにより、アプリケーションマネージャが、アクティブ状態に移行させる1つ以上のアプリケーションを選択できるようにするために、アプリケーションマネージャによって用いられてもよい。

【0039】

上述のインターフェースによって、アプリケーションマネージャとアプリケーションの両方が、アプリケーションライフサイクルを制御できる。インターフェースは明確に決まるのだが、図7および図8を参照して上述されたメソッドのボディは、関連するインターフェース（および関連する状態マシン）が共にコンパイルされている限りは、様々な方法で実装してよい。

【0040】

図9は、アプリケーションのライフサイクルの間に、アプリケーションマネージャとアプリケーションのいずれかが呼び出すことのできる代表的なメソッドを示す図である。図9に示すように、別個の一連のメソッドが、アプリケーションマネージャ902、およびアプリケーションマネージャ902によって管理されている各アプリケーション904によって呼び出されることが可能である。図7を参照して上述したように、第1の一連のメソッドは、アプリケーションマネージャ902によって呼び出されることが可能であり、`initialize`メソッド906、`start`メソッド908、`destroy`メソッド910、`pause`メソッド912を含む。同様に、図8を参照して上述したように、第2の一連のメソッドは、対応するアプリケーション904によって呼び出されることが可能であり、`paused`メソッド914、`destroyed`メソッド916、`resume request`メソッド918、`get property`メソッド920を含む。

【0041】

図10は、本発明の実施形態によるアプリケーションライフサイクルに従って、アプリケーションをロードおよび実行するために、アプリケーションマネージャを実装する1つの方法を示すプロセスのフローチャートである。そのプロセスは、ブロック1002で始まり、ブロック1004において、デジタルTV受信装置が起動される。次いで、`Java`環境がブロック1006において起動される。次いで、アプリケーションマネージャが、ブロック1008において構築され、1つ以上のアプリケーションのロードおよび実行を管理する。次いで、アプリケーションマネージャは、ブロック1010において実行される。

【0042】

図1に示したようなデジタルTV受信装置は、様々なデジタルテレビサービスを受信することが可能である。多数のチャンネル（すなわちサービス）を受信できるに加えて、これらのサービスは、対話型TVから、ビデオ・オン・デマンド、特殊プログラミングまでに及ぶだろう。より詳細には、テレビチャンネルによって提供されるサービスは、しばしば、音声、映像、アプリケーションを含むだろう。1012において（たとえば、ユーザがチャンネルを切り替えることにより）サービスが選択されると、ブロック1014において、選択されたサービスに関するデータストリームが、デジタルTV受信装置を介して受信される。たとえば、ユーザがディズニーのチャンネルに切り替えると、ディズニーのサービスが選択され、ディズニーのサービスに関連するデータストリームが受信される。

【0043】

ブロック1016において、選択されたサービスに関するアプリケーションが、データストリーム内に存在するか否かが決定される。たとえば、ユーザがディズニーのサービスを選択すると、ディズニーのアプリケーションを、データストリームの形で受信することができる。ブロック1018において、アプリケーションがデータストリーム内に存在することが決定されると、アプリケーションマネージャは、ブロック1022に示すようにアプリケーションをロードし、1024においてアプリケーションライフサイクルにしたがってアプリケーションを実行する。ロードおよび実行のプロセスに関しては、それぞれ図1.1および図1.2を参照してさらに詳細に説明する。プロセスは、ブロック1026に示すように完了する。データストリーム内にアプリケーションが存在しない場合には、プロセスは1020に戻り、アプリケーションに関する新しい信号がブロック1014で受信されたデータストリーム内に存在するのを待つ。次いで、プロセスは、ブロック1016において継続する。

【0044】

図1.1は、図10のブロック1022に示したようにアプリケーションをロードする1つの方法を示すプロセスのフローチャートである。プロセスは、ブロッ

ク1102で始まる。ブロック1104において、図3に示したような信号データが、アプリケーションマネージャによって受信される。信号データが受信されると、アプリケーションマネージャは、アプリケーション位置を特定し、ロードすることができる。アプリケーションに関する情報を格納するために、ブロック1106において、図4Bに示したようなアプリケーションコンテキストが作成される。次いで、アプリケーションコンテキストは、アプリケーションに関する全情報への参照を保持するために用いられる。たとえば、アプリケーションマネージャが、ブロック1108に示すようにアプリケーションのためのクラスローダを作成すると、クラスローダへの参照が、アプリケーションコンテキスト内に保持されてもよい。次いで、アプリケーションに関するクラスは、ブロック1110においてクラスローダによってロードされる。次いで、アプリケーションのインスタンスが、ブロック1112においてクラスから作成される。

【0045】

アプリケーションがインスタンス化されると、そのアプリケーションは、ブロック1114に示すようにロード済み状態に入る。アプリケーションが、その環境に関する属性を取得することと、アプリケーションマネージャと通信することを可能とするために、図8に示すようなアプリケーション環境インターフェースに従うアプリケーション環境オブジェクトが、ブロック1116においてインスタンス化される。次いで、アプリケーション環境オブジェクトは、ブロック1118においてinitializeメソッドがアプリケーションマネージャによって呼び出された際に、アプリケーションに対して初期化される。

【0046】

アプリケーションをロードすることに加えて、アプリケーションマネージャは、アプリケーションの実行の監視および管理を担う。図12は、図10のブロック1024に示したようにアプリケーションライフサイクルに従ってアプリケーションを実行する1つの方法を示すプロセスのフローチャートである。図7および図8に示すようなインターフェースを用いることにより、アプリケーションとアプリケーションマネージャは、アプリケーションの状態変更を提示、遅延、防止、もしくは達成することができる。以下において、アプリケーションの状態が

、アプリケーションのライフサイクルによって変更される方法を簡単に説明する。プロセスは、ブロック1202で始まる。アプリケーションがロード済み状態に入ると、アプリケーションは、ブロック1204においてアプリケーションマネージャがアプリケーション状態を一時停止状態に変更するのを待つ。たとえば、上述のように、アプリケーションマネージャは、アプリケーションインターフェースに備えられた `initialize` メソッドを用いてアプリケーションを初期化し、ブロック1206に示すようにアプリケーションを一時停止状態に入らせる。ブロック1206に関しては、図13を参照してさらに詳細に説明する。アプリケーションが一時停止状態に変更されると、アプリケーションマネージャは、その後ブロック1208において（たとえば、`start` メソッドを用いて）最終的にアプリケーション状態をアクティブ状態に変更する。ブロック1208に関しては、図14を参照してさらに詳細に説明する。アプリケーションをアクティブ状態に入らせることができるのはアプリケーションマネージャのみであることに注意すべきである。アクティブ状態からは、アプリケーションとアプリケーションマネージャのいずれでも、ブロック1210において、アプリケーション環境インターフェース内で利用可能な `paused` メソッドもしくはアプリケーションインターフェース内で利用可能な `pause` メソッドをそれぞれ用いることにより、アプリケーションを一時停止状態に入らせることができる。アプリケーションは、一時停止された場合、アプリケーション環境インターフェース内で利用可能な `resume request` メソッドにより実行の再開を要求してもよい。アプリケーションとアプリケーションマネージャのいずれでも、ブロック1212において、アプリケーション環境インターフェース内で利用可能な `destroyed` メソッドもしくはアプリケーションインターフェース内で利用可能な `destroy` メソッドをそれぞれ用いることにより、アプリケーションを終了し、抹消状態に入らせることができる。図示されているように、アプリケーションは、ブロック1212において、ロード済み、一時停止、アクティブ状態のいずれかから、抹消状態に入ることができる。たとえば、アプリケーションが実行を完了すると、そのアプリケーションは、抹消状態に入ってもよい。さらに、ロード、始動、終了のような処理の開始は、テレビ受信装置によって

受信された信号に応じてなされてよい。これは、たとえば、新しい番組が始まる場合、もしくはユーザがテレビのリモートコントロールのボタンを押して新しいチャンネルを選択した場合に生じる。アプリケーションが、要求のとおり状態を変更すると（たとえば、メソッドが適切に戻ると）、アプリケーションマネージャは、図4Aおよび図4Bに示すように、関連するアプリケーションコンテキスト内のアプリケーションの現在の状態を更新する。図12を参照して説明されていないが、上述の状態変更は、アプリケーションによって生成されたすべての状態変更例外にしたがって実行される。それゆえ、ほとんどの場合、状態変更例外がアプリケーションによって生成されると、アプリケーションマネージャは、アプリケーションの現在の状態を更新しない。その代わりに、アプリケーションマネージャは、後に、同じ状態変更、別の状態変更、もしくは別のアプリケーションの状態変更の要求を試みてもよい。

【0047】

抹消されると、アプリケーションによって保持されているリソースは開放されてよい。たとえば、アプリケーションのクリーンアップを開始するために、信号がアプリケーションマネージャによって（たとえば、受信装置を介してアプリケーションもしくは放送環境から）受信されてもよい。より詳細には、クラスローダは、ブロック1214においてアプリケーションに関するクラスをアンロードできる。さらに、図4Aおよび図4Bに示されているようにアプリケーションコンテキストリストによって参照されているアプリケーションコンテキストが開放されてもよい。

【0048】

上述のように、アプリケーションマネージャは、アプリケーション実行の準備をするために、アプリケーションの状態をロード済み状態から一時停止状態に変更する。図13は、図12のブロック1206に示したようにアプリケーションの状態をロード済みから一時停止に変更する1つの方法を示すプロセスのフローチャートである。プロセスは、ブロック1302で始まる。ブロック1304において、アプリケーションマネージャは、アプリケーションインターフェース内のinitializeメソッドを呼び出して、アプリケーション環境オブジェ

クトをパラメータとして渡す。次いで、アプリケーションは、ブロック1306においてアプリケーション環境オブジェクトを用いて自身を初期化する。たとえば、アプリケーション環境オブジェクトは、自身を初期化しているときに用いることのあるルートコンテナのような属性を取得してもよい。その後、アプリケーションは、ブロック1308に示すように一時停止状態に入る。

【0049】

一実施形態によると、アプリケーションの状態をアクティブ状態に変更できるのはアプリケーションマネージャのみである。図14は、図12のブロック1208に示したようにアプリケーションの状態を一時停止からアクティブに変更する1つの方法を示すプロセスのフローチャートである。プロセスは、ブロック1402で始まる。ブロック1404において、アプリケーションは、信号を受信し、アプリケーションを始動する。たとえば、そのような始動(`start`)信号は、新しい番組が始まる場合、もしくはユーザがテレビのリモートコントロールのボタンを押した際に、アプリケーションマネージャによって受信されてよい。サービスによっては、チャンネルが合わされた際に、アプリケーションが自動的に始動してもよい。次いで、アプリケーションマネージャは、ブロック1406に示すようにアプリケーションインターフェースに備えられた`start application`メソッドを呼び出す。アプリケーションは、ブロック1408において、サービスを実行してアクティブ状態に入る。

【0050】

本発明は、様々な場面で応用することができる。1つの例は、様々なチャンネル上でモニタ(たとえば、テレビスクリーン)の下方に株価を表示するというものである。株価情報は、中央サーバもしくは放送局に(たとえば、モデムケーブルもしくはインターネットへのその他の接続を介して)つながる放送サービスもしくは裏チャンネルから受信装置によって取得されることが可能である。株価表示アプリケーションが、初期化後に一時停止状態になると仮定すると、ユーザは、アプリケーションマネージャが株価表示アプリケーションを始動するよう信号通知するテレビのリモコンのボタンを押す。アプリケーションマネージャは、株価表示アプリケーションのための`start application`メソッド

を呼び出す。アプリケーションマネージャは、この時点で、アプリケーションがサービスを実行しているとみなす。株価表示アプリケーションは、始動信号を受信する際に、株価を検索するために裏チャンネルを開く新しいスレッドを作成する。これで、株価表示アプリケーションはアクティブ状態となる。

【0051】

アクティブ状態の間、株価表示アプリケーションは、株価を表示し続ける。しかしながら、アプリケーションの制御を超える状況が生じると、株価表示アプリケーションが、株価の更新を検索できなくなる可能性もある。これらの状況下では、アプリケーションは、利用可能な最近の株価を表示し続けるよう決定してもよい。しかしながら、ある期間が経っても、アプリケーションが裏チャンネルを開くことができない場合もある。したがって、アプリケーションは、表示している株価があまりに古いため、サービスをもはや実行できないと決定してもよい。次いで、アプリケーションは、アプリケーション環境オブジェクト上の `pause` メソッドを呼び出すことによりアクティブ状態から抜け出すよう選択し、この変更をアプリケーションマネージャに信号通知することができる。さらに、アプリケーションは、この時点もしくは後の時点で、もはやサービスを実行する機会がないと決定してもよく、その場合は、終了を決定する。アプリケーションは、いくつかのクリーンアップを実行し、もはや必要としないリソースを開放する。次いで、アプリケーションは、終了してアプリケーション環境オブジェクト上の `destroyed` メソッドを呼び出し、抹消状態に入ったことをアプリケーションマネージャに信号通知する。次いで、アプリケーションマネージャは、対応するクリーンアップをこのアプリケーションに対して実行する。

【0052】

本発明は、任意の適切なコンピュータシステムに実装することができる。図15は、本発明の実装に適した代表的な汎用コンピュータシステム1502を示している。コンピュータシステムは、任意の適切な形態をとってよい。たとえば、コンピュータシステムは、デジタルTV受信装置もしくはセットトップボックスと統合されてもよい。

【0053】

当業者の理解するように、コンピュータシステム1530、すなわち、より詳細にはCPU1532は、仮想マシンをサポートするよう構成されてもよい。コンピュータシステム1502は、一次記憶装置1506（一般には読み取り専用メモリすなわちROM）と一次記憶装置1508（一般にはランダムアクセスメモリすなわちRAM）とを含むメモリ装置に接続可能な任意の数のプロセッサ1504（中央演算処理装置すなわちCPUとも呼ばれる）を備える。当業者に周知のとおり、ROMは、データや命令を単方向的にCPU1504へ転送する機能を持ち、RAMは、一般に、データや命令を双方向的に転送するために用いられる。一次記憶装置1506および1508は両方とも、任意の適切なコンピュータ読み取り可能な媒体を含んでよい。CPU1504は一般に、任意の数のプロセッサを含んでいてよい。

【0054】

データ記憶容量を追加するために、二次記憶媒体1510（一般にはマスメモリ装置）が、双方向的にCPU1504に接続されてもよい。マスメモリ装置1510は、コンピュータコード、データなどを含むプログラムを格納するために用いることのできるコンピュータ読み取り可能な媒体である。一般に、マスメモリ装置1510は、一次記憶装置1506および1508よりも一般に低速であるハードディスクのような記憶媒体である。

【0055】

CPU1504は、さらに、1つ以上の入出力装置1512に接続されてもよい。入出力装置1512は、ビデオモニタ、トラックボール、マウス、キーボード、マイクロホン、タッチセンシティブ・ディスプレイ、トランスデューサ・カード読み取り装置、磁気もしくは紙テープ読取装置、タブレット、スタイラス、音声もしくは手書き文字認識装置、もしくは、他のコンピュータのような周知の入力装置などの装置を含むが、それらに限定されない。最後に、CPU1504は随意的に、1514に一般的に示したネットワーク接続を用いて、コンピュータもしくは遠隔通信ネットワーク（たとえばインターネットネットワークもしくはイントラネットネットワーク）に接続されてもよい。そのようなネットワーク接続を行った場合、CPU1504は、上述した方法のステップの実行中に、ネ

ットワークから情報を受信し、ネットワークに情報を出力することができると考えられる。そのような情報は、CPU1504を用いて実行される一連の命令として表されることが多く、たとえば搬送波で実現されるコンピュータデータ信号の形で、ネットワークから受信され、ネットワークに出力されることが可能である。

【0056】

アプリケーションライフサイクルに従って各アプリケーションをロードし、その実行を管理するためにアプリケーションマネージャを用いることにより、アプリケーションの実行が標準化され、メモリリソースが効果的に温存される。アプリケーションライフサイクルは、部分的には、アプリケーションが入ることのできる状態を一貫した方法で制御することにより実装される。一実施形態によると、アプリケーションライフサイクルの制御に適切な方法を定義するインターフェースが、アプリケーションマネージャだけでなくアプリケーションにも備えられている。このように、アプリケーションとアプリケーションマネージャは、状態変更、状態変更の要求、状態変更を遅延もしくは取り消しする要求を、その他の状態に関する情報と共に通信してもよい。本発明を用いることにより、アプリケーションは、異なる時間に異なるマシン上で作動されるが、予測どおりの同じ結果を生成する。さらに、本発明は、もはや必要ではないと決定された際に、アプリケーションに関するメモリが開放されることを可能とする。デジタル受信装置やセットトップボックスのメモリリソースに限られているとすれば、デジタルテレビ環境のメモリリソースを管理する能力は、著しい改善と言える。

【0057】

本発明の実施形態と応用を示して説明したが、本発明の概念、範囲、意図の範囲内で多くの変形と修正が可能であり、これらの変形は、本明細書を精読すれば当業者には明らかとなるだろう。たとえば、本発明は、デジタルTV受信装置に実装した例によって説明されている。しかしながら、本発明は、他の状況で用いられてもよい。さらに、本発明は、JAVAプラットフォームに実装した例によって説明されているが、別のプラットフォームに実装されてもよい。さらに、上述のプロセスのブロックは、例示に過ぎない。したがって、アプリケーションマ

ネージャとアプリケーションライフサイクルの実装は、代替のデータ構造はもちろん、代替のプロセスのブロックを用いて実行されてもよい。さらに、アプリケーションマネージャとアプリケーションは、別個のインターフェースを持つように説明されているが、これらのインターフェースは、アプリケーションだけでなく全アプリケーションにも可視であるpublicメソッドを含んでもよい。したがって、本実施形態は、例示的なものであって制限的ではないとみなされ、本発明は、本明細書の詳細に限定されず、添付の請求項の範囲と等価物の範囲内で修正することができる。

【図面の簡単な説明】

【図1】

本発明を実装可能なデジタルTV受信装置を示すブロック図である。

【図2A】

デジタルTV受信装置を通してロードされたアプリケーションを管理するために、アプリケーションマネージャが実装された本発明の一実施形態を示すブロック図である。

【図2B】

図2Aのアプリケーションマネージャが動作の際に従うことのできる代表的な一連のルールを示す図である。

【図3】

図2Aのアプリケーションマネージャによって受信された信号データを格納するために用いられる代表的なデータ構造を示すブロック図である。

【図4A】

関連するアプリケーションの管理および実行中に、アプリケーションマネージャによってアクセスされる代表的なコンテキストリストを示す図である。

【図4B】

代表的なコンテキストリスト内で識別されたアプリケーションコンテキストを格納するために用いられる代表的なデータ構造を示す図である。

【図5A】

本発明の一実施形態に従って、ディスプレイマネージャによってアクセスされ

るディスプレイコンテキストの代表的なリストを示す図である。

【図5B】

ディスプレイコンテキストに関連する代表的な状態図である。

【図5C】

アプリケーションマネージャが従う一連の代表的なルールに従って生成された一連の代表的な表示状態を示す図である。

【図6】

アプリケーションのライフサイクルの間にアプリケーションが入る一連の状態を示す状態図である。

【図7】

アプリケーションのライフサイクルの間にアプリケーションマネージャが呼び出すことのできるメソッドを識別する代表的なアプリケーションインターフェースを示す図である。

【図8】

アプリケーションのライフサイクルの間にアプリケーションが呼び出すことのできるメソッドを識別する代表的なアプリケーション環境インターフェースを示す図である。

【図9】

アプリケーションのライフサイクルの間にアプリケーションマネージャとアプリケーションのいずれかが呼び出すことのできるメソッドを示す図である。

【図10】

本発明の実施形態によるアプリケーションライフサイクルに従って、アプリケーションをロードし実行するために、アプリケーションマネージャを実装する1つの方法を示すプロセスのフローチャートである。

【図11】

図10のブロック1022に示したようにアプリケーションをロードする1つの方法を示すプロセスのフローチャートである。

【図12】

図10のブロック1024に示したようにアプリケーションライフサイクルに

従ってアプリケーションを実行する1つの方法を示すプロセスのフローチャートである。

【図13】

図12のブロック1206に示したようにアプリケーションの状態をロード済みから一時停止に変更する1つの方法を示すプロセスのフローチャートである。

【図14】

図12のブロック1208に示したようにアプリケーションの状態を一時停止からアクティブに変更する1つの方法を示すプロセスのフローチャートである。

【図15】

本発明の実装に適した代表的な汎用コンピュータシステムを示すブロック図である。

【符合の説明】

- 102 アンテナ
- 104 チューナモジュール
- 106 MPEG2転送ストリーム
- 108 多重分離装置
- 110 暗号化MPEGストリーム
- 112 映像ストリーム
- 114 音声ストリーム
- 116 データストリーム
- 118 条件付きアクセスサブシステム
- 120 キー管理情報
- 122 復号情報
- 123 復号化MPEGストリーム
- 124 映像ストリーム
- 125 音声ストリーム
- 126 データ
- 128 デコーダ
- 130 フレームバッファ

- 132 スピーカ
- 200 デジタルTV受信装置
- 202 放送データストリーム
- 204 信号モニタ
- 206 信号データ
- 208 アプリケーションマネージャ
- 210 ディスプレイコンテキスト
- 212 ディスプレイマネージャ
- 214 テレビモニタ
- 216 ルール
- 218 第1のルール
- 220 第2のルール
- 222 第3のルール
- 224 第4のルール
- 302 信号データ
- 304 アプリケーション存在の指示
- 306 ロケーション
- 308 認証情報
- 310 許可
- 400 アプリケーション
- 400-A アプリケーション
- 400-B アプリケーション
- 400-C アプリケーション
- 400-D アプリケーション
- 402 アプリケーションコンテキストリスト
- 404 アプリケーションコンテキスト
- 404-A アプリケーションコンテキスト
- 404-B アプリケーションコンテキスト
- 404-C アプリケーションコンテキスト

404 - D アプリケーションコンテキスト
406 クラスローダID
408 信号データID
410 ディスプレイコンテキストID
412 アプリケーションID
414 アプリケーション環境オブジェクト
416 現在のアプリケーション状態
502 ディスプレイコンテキストリスト
504 コンテキストID
506 ディスプレイコンテキスト
508 可視
510 不可視
512 テーブル
514 表示状態
516 アプリケーション状態
602 ロード状態
604 一時停止状態
606 アクティブ状態
702 アプリケーションインターフェース
704 initializeメソッド
706 アプリケーション環境オブジェクト
708 startメソッド
710 pauseメソッド
712 destroyメソッド
714 パラメータ
802 アプリケーション環境インターフェース
804 destroyedメソッド
806 pausedメソッド
808 get propertyメソッド

810 パラメータ
812 resume requestメソッド
902 アプリケーションマネージャ
904 アプリケーション
906 initializeメソッド
908 startメソッド
910 destroyメソッド
912 pauseメソッド
1502 コンピュータシステム
1504 プロセッサ、CPU
1506 一次記憶装置
1508 一次記憶装置
1510 二次記憶装置、マスメモリ装置
1512 入出力装置
1514 ネットワーク
1530 コンピュータシステム
1532 CPU
1514 ネットワーク

【図1】

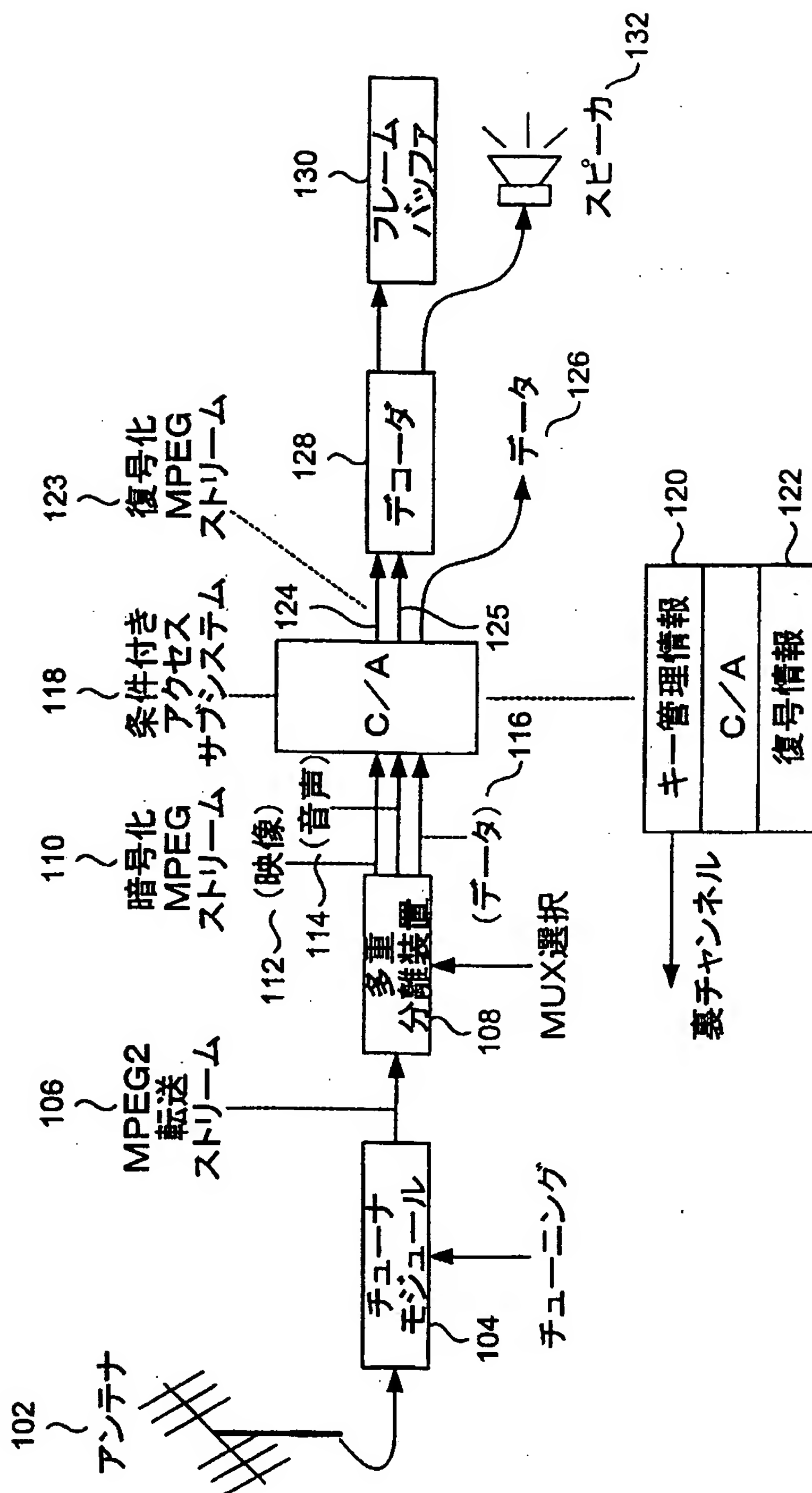


FIG. 1

【図2A】

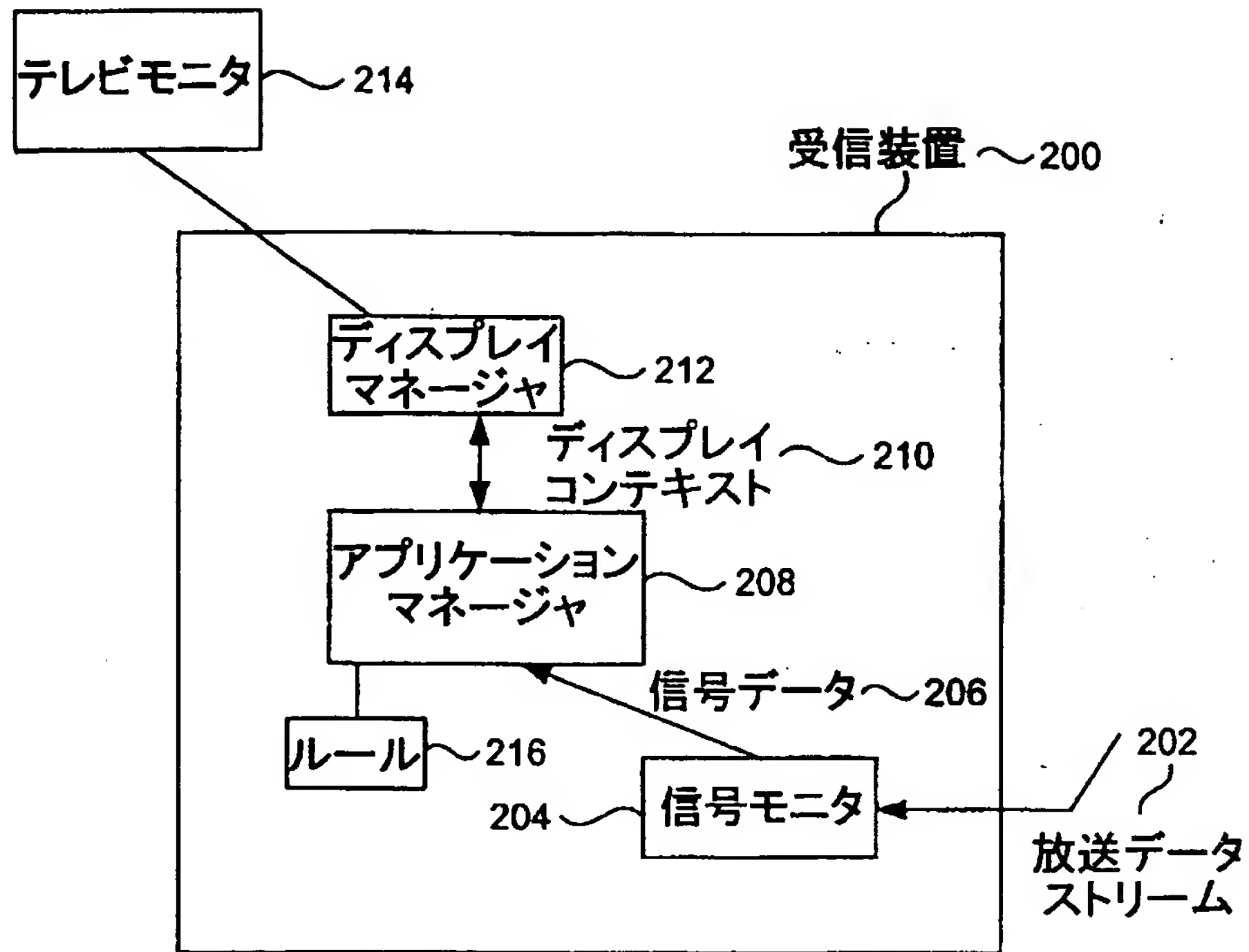


FIG. 2A

【図2B】

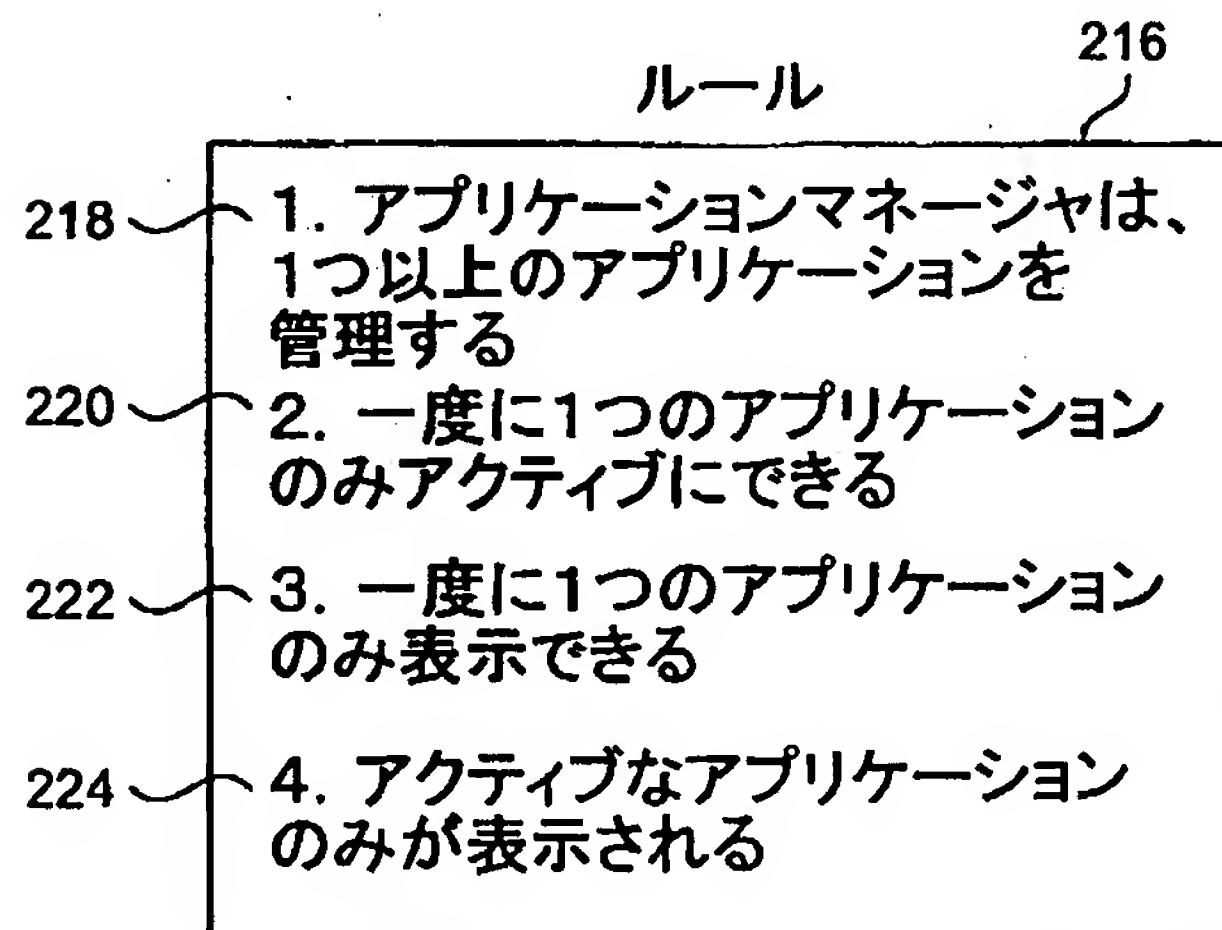


FIG. 2B

【図3】

信号データ ～ 302

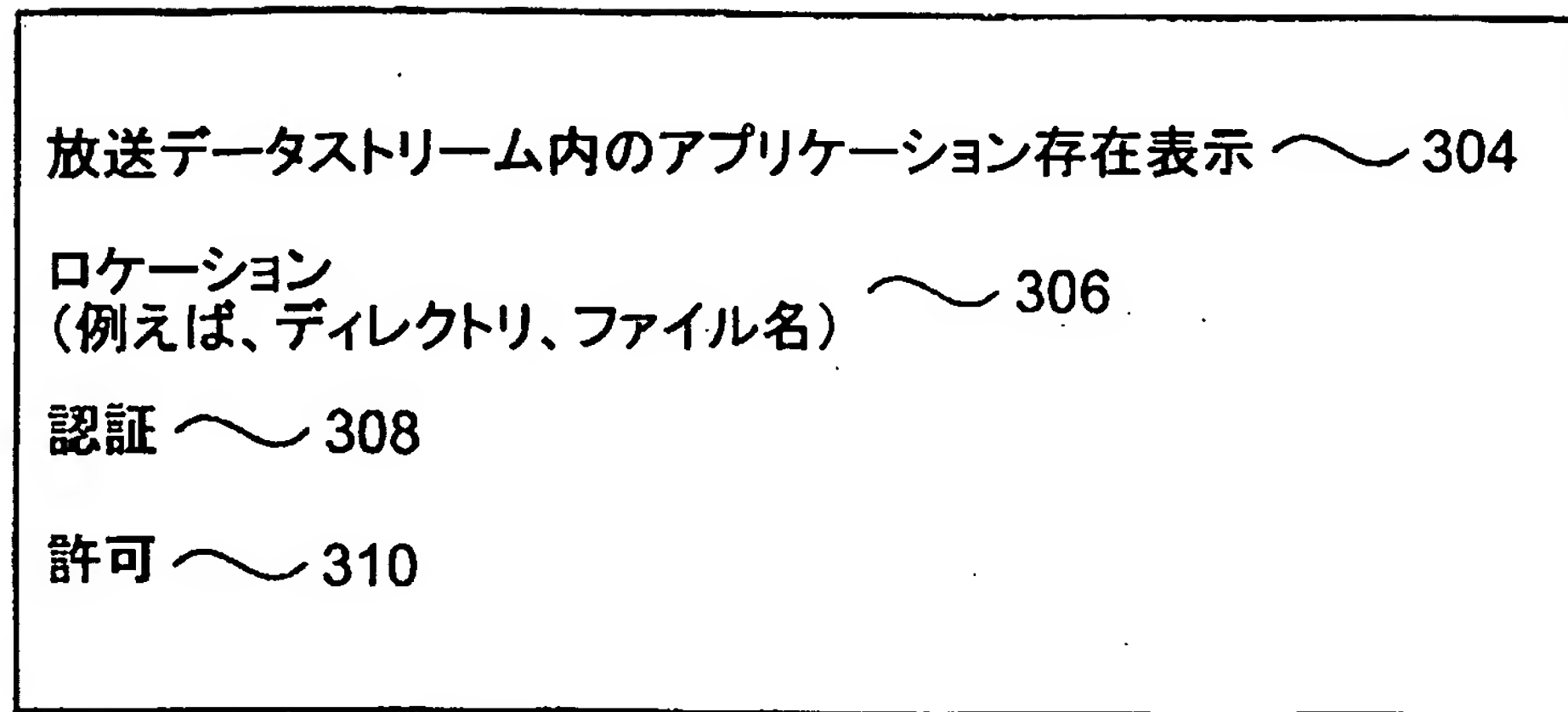


FIG. 3

【図4A】

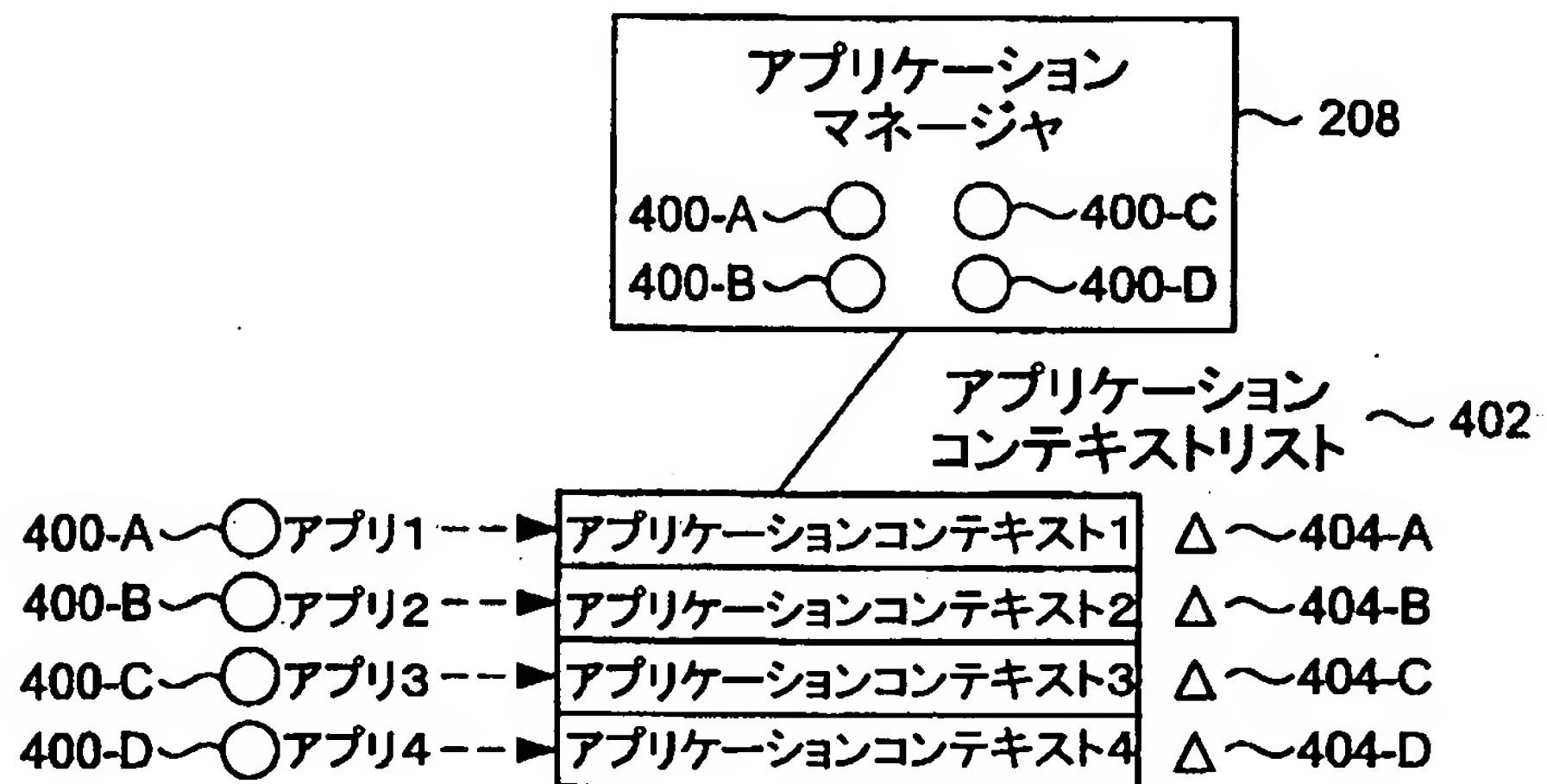


FIG. 4A

【図4B】

アプリケーションコンテキストΔ～404

クロスローダID	406
信号データID	408
ディスプレイコンテキストID	410
アプリケーションID	412
アプリケーション環境オブジェクト	414
現在のアプリケーション状態	416



FIG. 4B

【図5A】

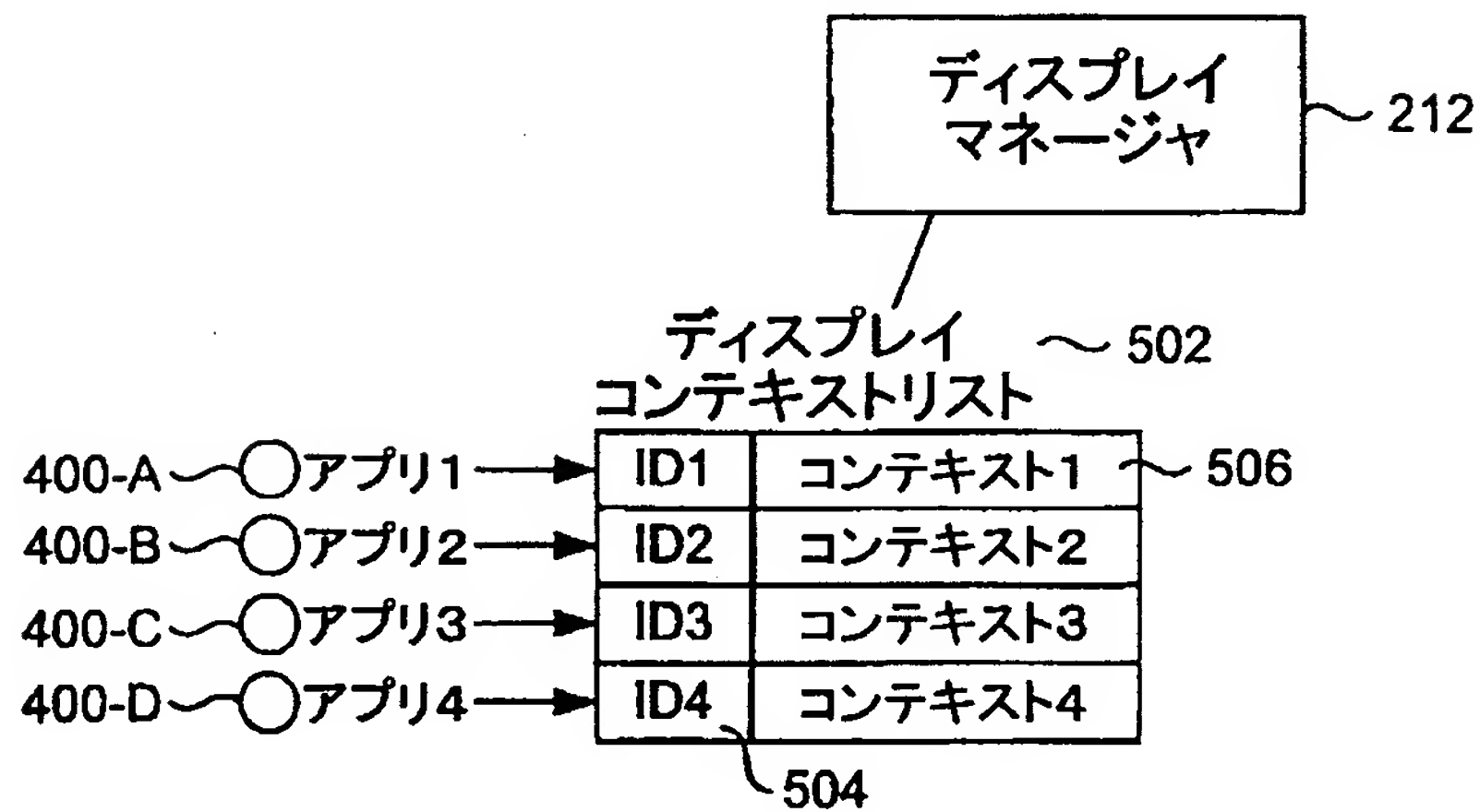


FIG. 5A

【図5B】

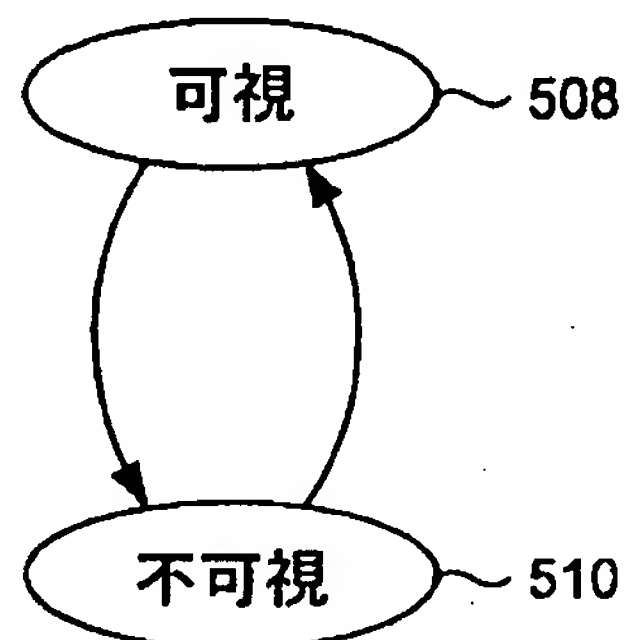


FIG. 5B

【図5C】

512 ↙

		0	1
516	アプリケーション状態	アクティブ	ポーズド
514	表示状態	可視	不可視

FIG. 5C

【図6】

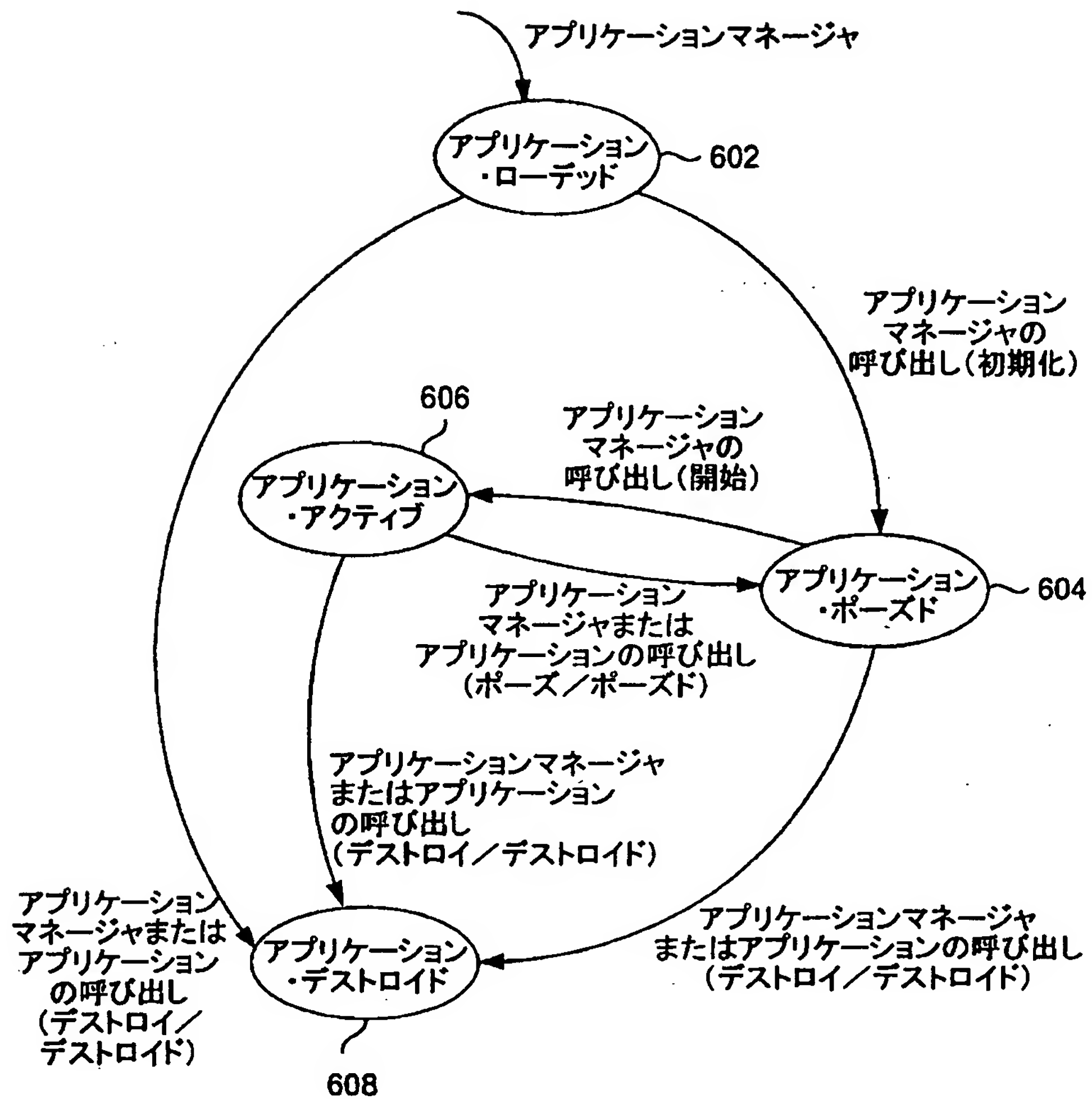


FIG. 6

【図7】

アプリケーションプログラミング
インターフェース(API)

702
Interface application {
704 ~ public void init application (Application environment env);
708 ~ public void start application ();
710 ~ public void pause application();
712 ~ public void destroy application (boolean unconditional);
714
}

FIG. 7

【図8】

アプリケーション環境インターフェース

802
Interface Application_environment {
804 ~ public void destroyed ();
808 ~ public java.lang.object.get property (string key);
806 ~ public void paused ();
812 ~ public void resume_request ();
810
}

FIG. 8

【図9】

アプリケーション・マネージャ ～ 902

アプリケーション ～ 904

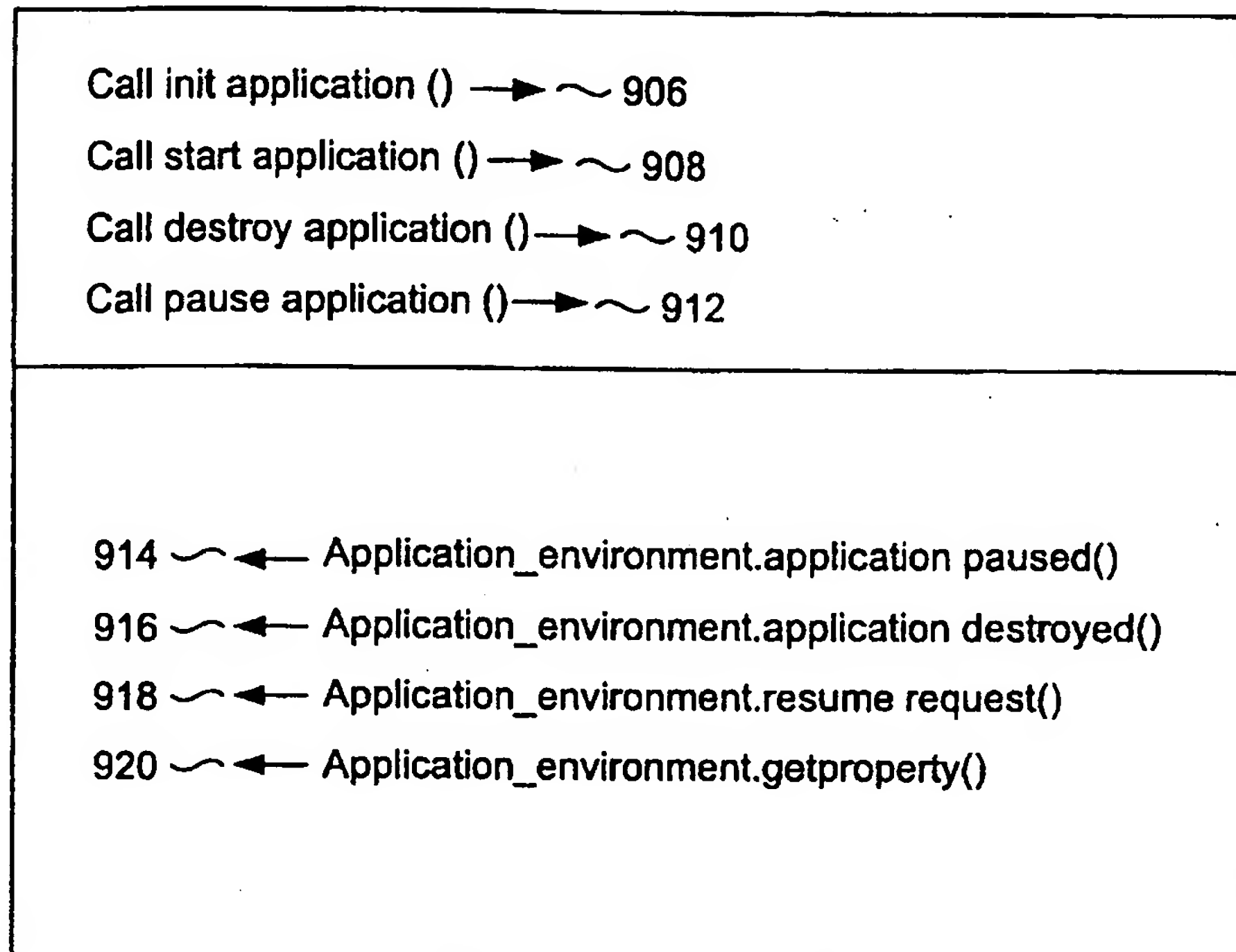


FIG. 9

【図10】

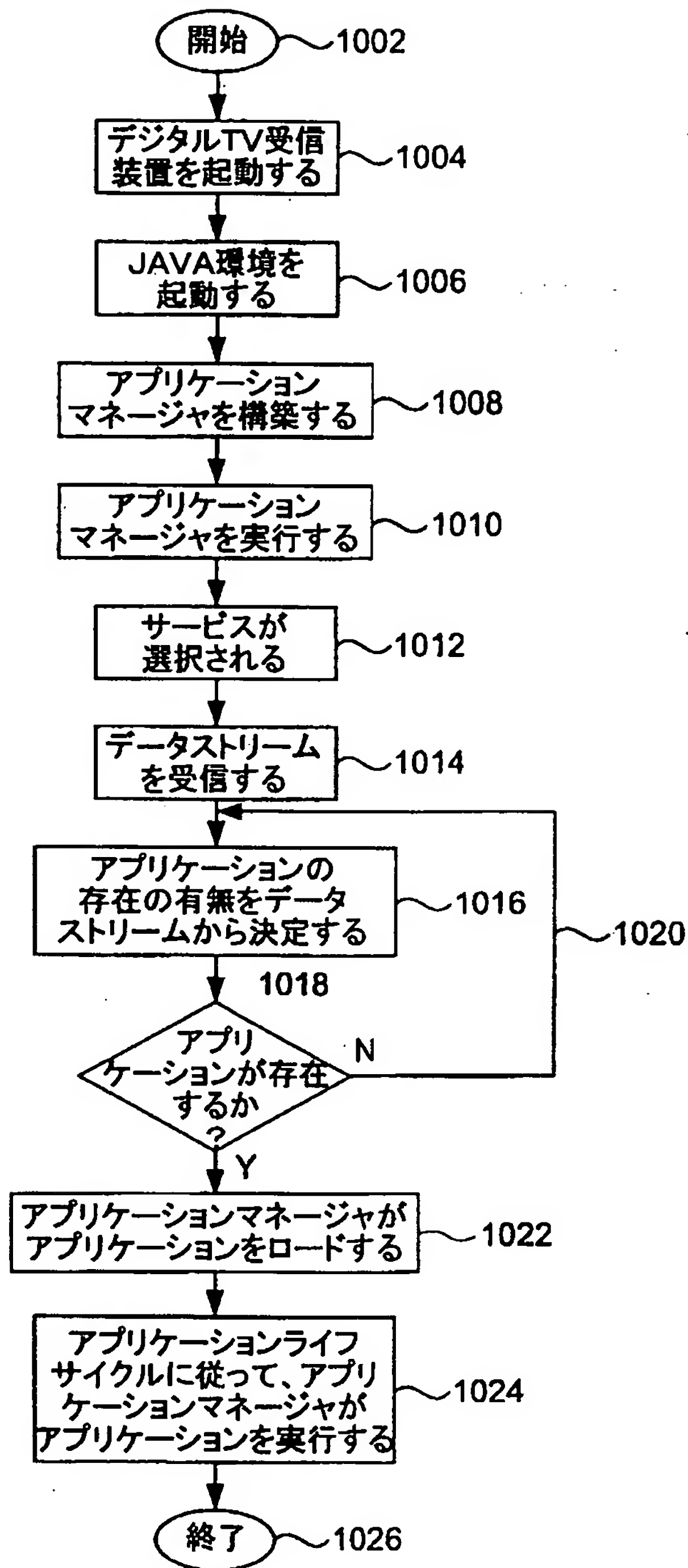


FIG. 10

【図11】

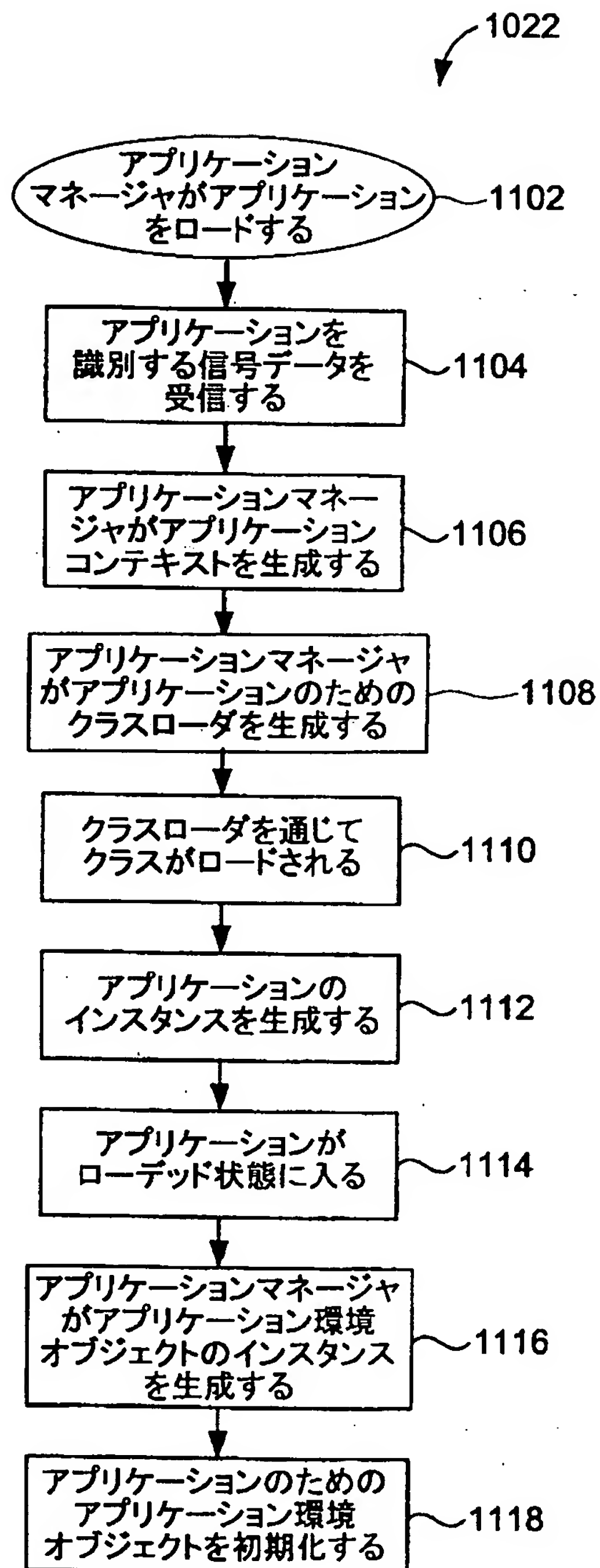


FIG. 11

【図12】

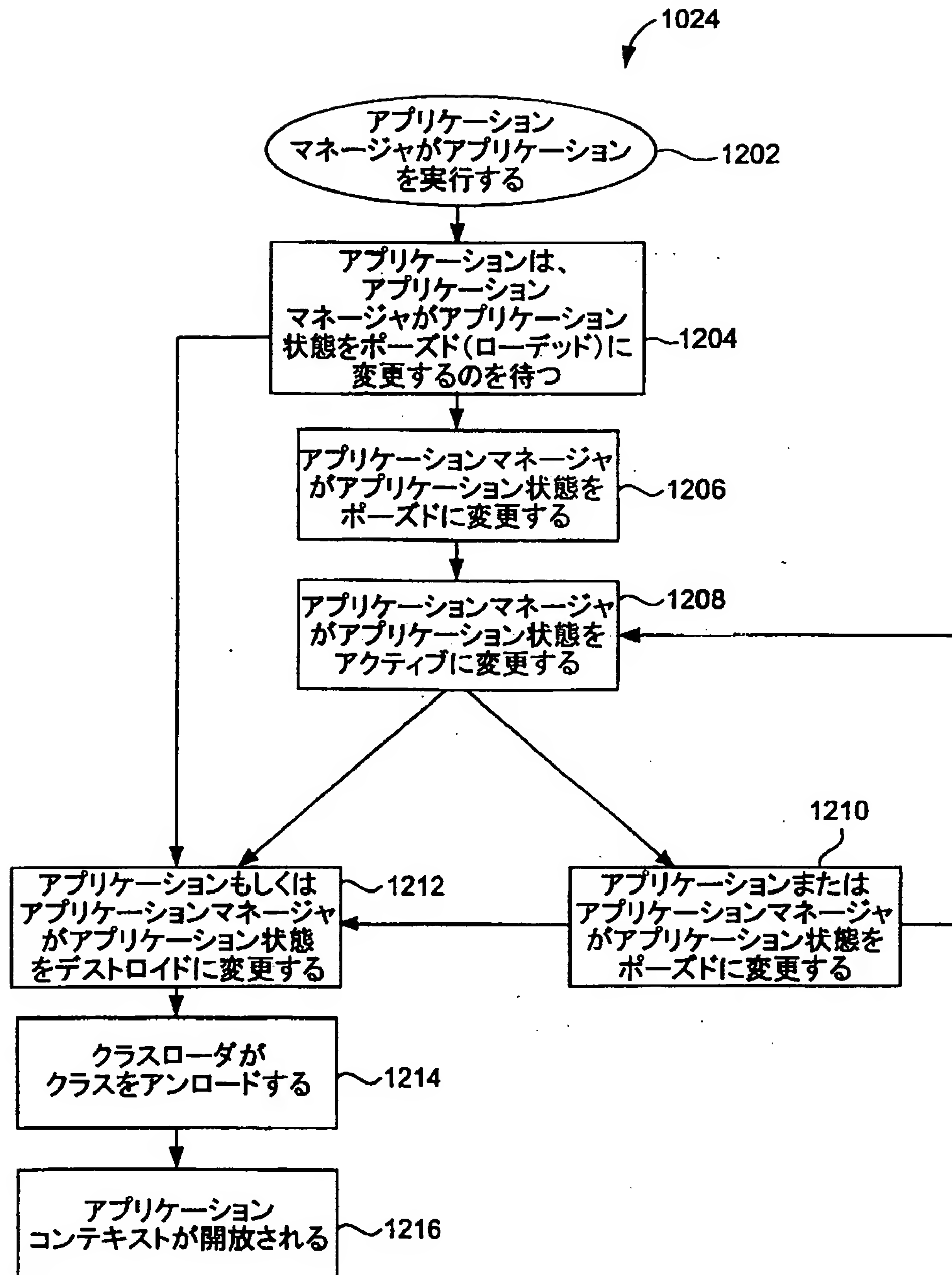


FIG. 12

【図13】

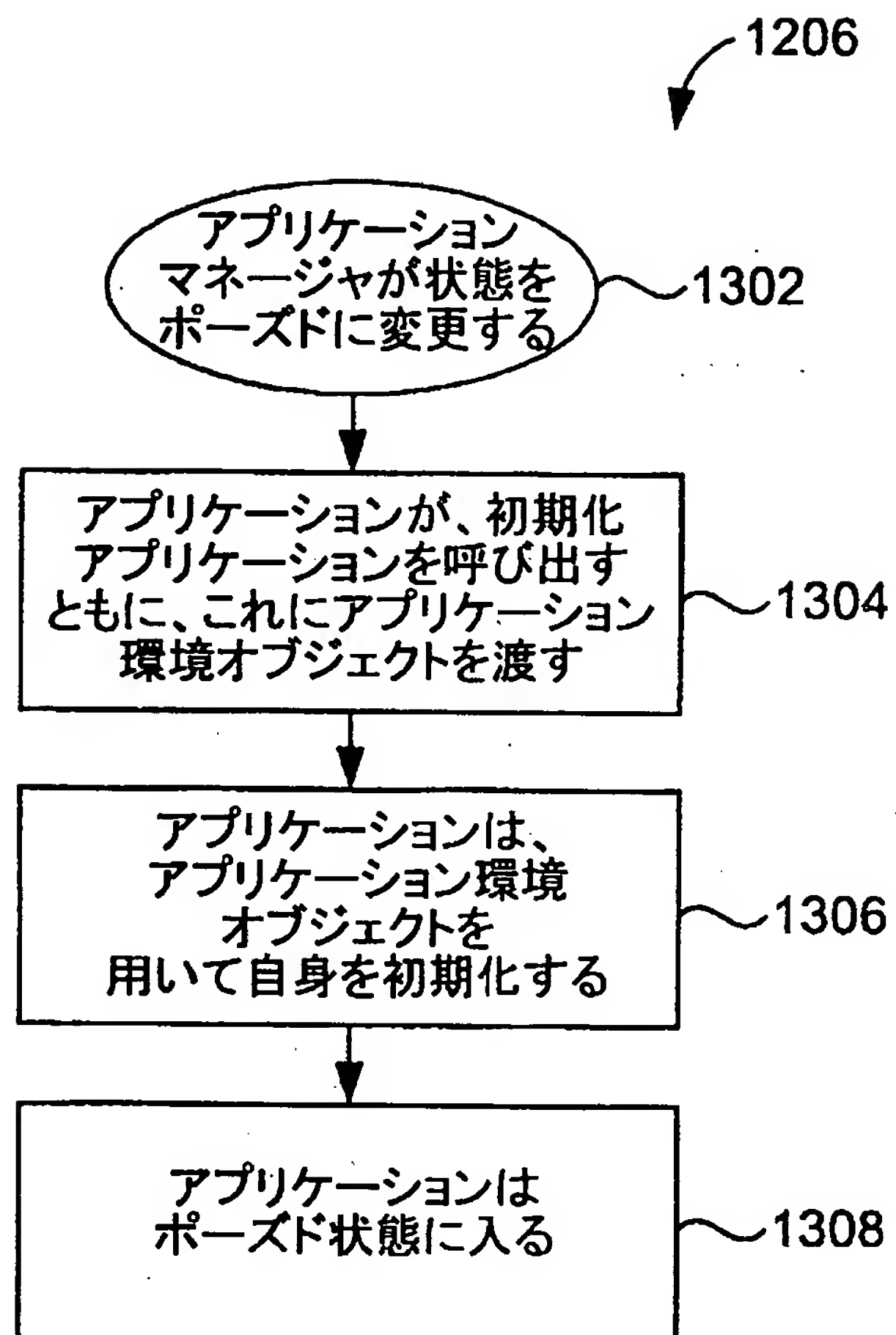


FIG. 13

【図14】

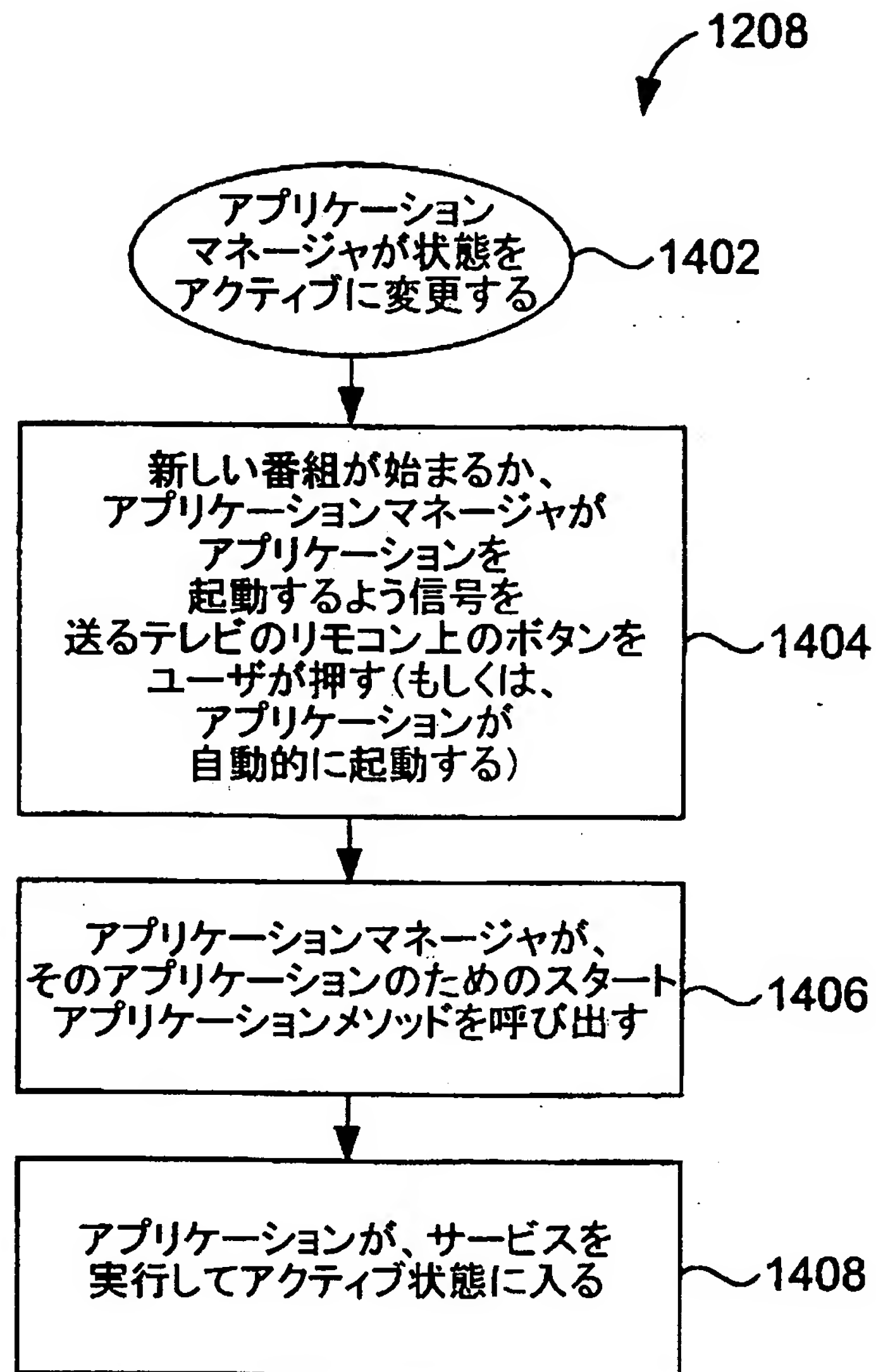


FIG. 14

【図15】

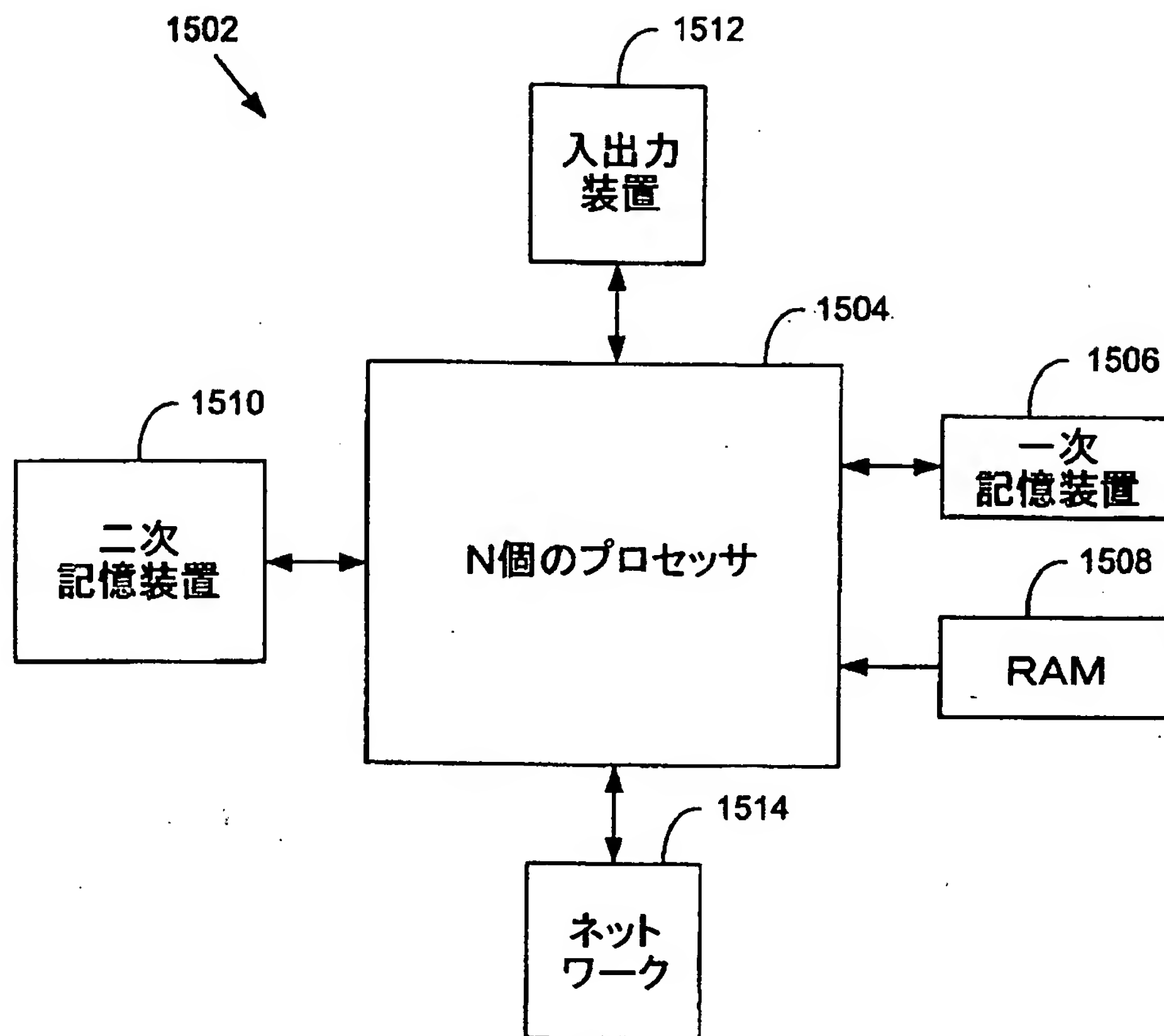


FIG. 15

【国際調査報告】

INTERNATIONAL SEARCH REPORT

International Application No. PC1/US 00/19167	
A. CLASSIFICATION OF SUBJECT MATTER IPC 7 G06F9/46 G06F9/445	
According to International Patent Classification (IPC) or to both national classification and IPC	
B. FIELDS SEARCHED Minimum documentation searched (classification system followed by classification symbols) IPC 7 G06F	
Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched	
Electronic data base consulted during the international search (name of data base and, where practical, search terms used) EPO-Internal	
C. DOCUMENTS CONSIDERED TO BE RELEVANT	
Category *	Citation of document, with indication, where appropriate, of the relevant passages Relevant to claim No.
X	EP 0 524 071 A (BULL SA) 20 January 1993 (1993-01-20)
A	column 9, line 40 -column 10, line 27 column 11, line 16 -column 12, line 5 column 14, line 57 -column 16, line 38
X	EP 0 908 821 A (CANAL+) 14 April 1999 (1999-04-14)
A	column 3, line 35 -column 11, line 50
	EP 0 333 617 A (IBM) 20 September 1989 (1989-09-20) column 6, line 35 -column 11, line 57
	-/-
<input checked="" type="checkbox"/> Further documents are listed in the continuation of box C.	
<input checked="" type="checkbox"/> Patent family members are listed in annex.	
* Special categories of cited documents: *A* document defining the general state of the art which is not considered to be of particular relevance *E* earlier document but published on or after the international filing date *L* document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified) *O* document referring to an oral disclosure, use, exhibition or other means *P* document published prior to the international filing date but later than the priority date claimed *T* later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention *X* document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone *Y* document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art *S* document member of the same patent family	
Date of the actual completion of the international search 20 June 2001	Date of mailing of the international search report 02/08/2001
Name and mailing address of the ISA European Patent Office, P.B. 6818 Patentlaan 2 NL - 2260 HV Rijswijk Tel. (+31-70) 340-2040, Tx. 31 651 epo nl, Fax (+31-70) 340-3016	Authorized officer Bijl, K

Form PCT/ISA/210 (second sheet) (July 1992)

INTERNATIONAL SEARCH REPORT

Inter. Patent Application No.
PCT/US 00/19167

C. (Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT

Category	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	BUNT, R.B.: "Scheduling Techniques for Operating Systems" COMPUTER, October 1976 (1976-10), pages 10-17, XP000199223 page 11, left-hand column, line 49 -page 12, left-hand column, line 37	1-75
A	THOM R: "MULTITASKING FUER MIKROCONTROLLER. \PARALLELVERARBEITUNG FUER SEQUENTIELLE SYSTEME MIT OS-51" ELEKTRONIK, DE, FRANZIS VERLAG GMBH. MUNCHEN, vol. 39, no. 15, 20 July 1990 (1990-07-20), pages 50-56, XP000138192 ISSN: 0013-5658 page 54, right-hand column, line 6 -page 55, right-hand column, line 11	1-75

INTERNATIONAL SEARCH REPORT

Information on patent family members

International Application No.

PCT/US 00/19167

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
EP 0524071 A	20-01-1993	FR 2679351 A	22-01-1993
		JP 2075132 C	25-07-1996
		JP 5274234 A	22-10-1993
		JP 7099511 B	25-10-1995
		US 5455950 A	03-10-1995
EP 908821 A	14-04-1999	AU 9363298 A	27-04-1999
		BR 9813024 A	15-08-2000
		CN 1281613 T	24-01-2001
		EP 1021918 A	26-07-2000
		HR 20000166 A	31-10-2000
		HU 0100296 A	28-05-2001
		WO 9918730 A	15-04-1999
		NO 20001652 A	07-06-2000
		PL 340331 A	29-01-2001
		TR 200000922 T	21-07-2000
EP 333617 A	20-09-1989	US 4912628 A	27-03-1990
		DE 68922769 D	29-06-1995
		DE 68922769 T	30-11-1995
		JP 1241631 A	26-09-1989
		JP 1961720 C	25-08-1995
		JP 6064538 B	22-08-1994

フロントページの続き

(31)優先権主張番号 09/465, 980

(32)優先日 平成11年12月16日(1999. 12. 16)

(33)優先権主張国 米国 (US)

(81)指定国 EP(AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OA(BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG), AP(GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZW), EA(AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CR, CU, CZ, DE, DK, DM, DZ, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, TZ, UA, UG, UZ, VN, YU, ZA, ZW

(72)発明者 イエ・タオ

アメリカ合衆国 カリフォルニア州94087
サニーベイル, #116, イースト・エル
カミノ・リアル, 929

(72)発明者 リバス・ジーザス・デイビッド

アメリカ合衆国 カリフォルニア州94131
サン・フランシスコ, 28番・ストリート, 258

(72)発明者 コートニー・ジョナサン・ディ.

アメリカ合衆国 カリフォルニア州95125
サン・ホセ, コーチ・レーン, 1623

(72)発明者 メッサー・キース・エル.

アメリカ合衆国 カリフォルニア州94541
ヘイワード, プリンストン・ストリート, 21806

Fターム(参考) 5B076 BA05 BB06

5B098 GA04 GC01 GD04 GD22

5C025 BA27 DA01 DA05